METHODOLOGY FOR SENSITIVITY ANALYSIS, APPROXIMATE ANALYSIS, AND DESIGN OPTIMIZATION IN CFD FOR MULTIDISCIPLINARY APPLICATIONS

By

Arthur C. Taylor III, Principal Investigator
Gene W. Hou, Co-Principal Investigator

Final Report
For the period ended December 31, 1995

August 1996

Old Dominion University Research Foundation

DEPARTMENT OF MECHANICAL ENGINEERING
COLLEGE OF ENGINEERING & TECHNOLOGY
OLD DOMINION UNIVERSITY
NORFOLK, VIRGINIA 23529

METHODOLOGY FOR SENSITIVITY ANALYSIS, APPROXIMATE ANALYSIS, AND DESIGN OPTIMIZATION IN CFD FOR MULTIDISCIPLINARY APPLICATIONS

By

Arthur C. Taylor III, Principal Investigator
Gene W. Hou, Co-Principal Investigator

Final Report
For the period ended December 31, 1995

August 1996

Overview

This final report, for grant NAG-1-1265, represents also the final approved Ph.D. dissertation of Dr. Vamshi Mohan Korivi. Although many additional things were accomplished through the several-year span of this grant, this report represents well the culmination of effort and accomplishments of this multi-year project.

Dr. Arthur C. Taylor III
Associate Professor

ABSTRACT

AERODYNAMIC DESIGN OPTIMIZATION WITH CONSISTENTLY DISCRETE SENSITIVITY DERIVATIVES VIA THE INCREMENTAL ITERATIVE METHOD

by
Vamshi M. Korivi
Old Dominion University, 1995
Director: Dr. A. C. Taylor III

An incremental iterative formulation together with the well-known spatially split approximate-factorization algorithm, is presented for solving the large, sparse systems of linear equations that are associated with aerodynamic sensitivity analysis. This formulation is also known as the "delta" or "correction" form. For the smaller two dimensional problems, a direct method can be applied to solve these linear equations in either the standard or the incremental form, in which case the two are equivalent. However, iterative methods are needed for larger two-dimensional and three dimensional applications because direct methods require more computer memory than is currently available. Iterative methods for solving these equations in the standard form are generally unsatisfactory due to an ill-conditioned coefficient matrix; this problem is overcome when these equations are cast in the incremental form. The methodology is successfully implemented and tested using an upwind cell-centered finite-volume formulation applied in two dimensions to the thin-layer Navier-Stokes equations for external flow over an airfoil. In three dimensions this methodology is demonstrated with a marching-solution algorithm for the Euler equations to calculate supersonic flow over the High-Speed Civil Transport configuration (HSCT 24E). The sensitivity derivatives obtained with

the incremental iterative method from a marching Euler code are used in a design-improvement study of the HSCT configuration that involves thickness, camber, and planform design variables.

ACKNOWLEDGMENTS

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

a	local speed of sound
C	chord
C_D	drag coefficient
C_L	lift coefficient
C_M	pitching-moment coefficient
C_x, C_y, C_z	force coefficients in x,y,z directions
$C_{m_x}, C_{m_y}, C_{m_z}$	moment coefficients in x,y,z directions
e	total energy per unit volume
$\widehat{F}, \widehat{G}, \widehat{H}$	inviscid fluxes in curvilinear coordinates
$\widehat{G_v}^{tl}$	viscous fluxes in curvilinear coordinates
i, j, k	nodal points/indices
J	Jacobian matrix
M_∞	free-stream Mach number
p	pressure
Q	field variables
R	residual vector
Re	Reynolds number
T	temperature
t	time
u, v, w	velocity components
x, y, z	cartesian coordinates
$\bar{X}$	vector of grid coordinates

Greek Symbols

α	angle of attack

$\bar{\beta}$	design variable vector
γ	ratio of specific heats
δ	finite difference operator
ϵ	convergence criterion
θ	polar angle
κ	spatial accuracy parameter
λ	Lagrange multiplier
μ	coefficient of viscosity
ρ	density
ξ, η, ζ	body-fitted coordinates

Subscripts:

B	boundary point
IP	interior point
∞	free-stream
x, y, z	co-ordinate directions

Superscripts:

m	iteration index
n	time iteration index
T	transpose
*	steady-state

Derivative Quantities

C_j'	sensitivity of the jth system response with respect to β_k
Q'	sensitivity of field variables with respect to β_k
$\left[\widetilde{\frac{\partial R}{\partial Q}}\right]$	Approximate Jacobian operator
$\left[\frac{\partial R}{\partial Q}\right], \left[\frac{\partial R}{\partial X}\right]$	Jacobian matrices
$\frac{\partial \bar{X}}{\partial \beta_k}$	grid sensitivity

Miscellaneous

∂	partial derivative
$\triangle$	backward difference operator
∇	forward difference operator

Chapter 1

INTRODUCTION

Rapid advances in computer technology have enabled fluid-flow simulations around full aircraft configurations with computational fluid dynamics (CFD). Numerical simulation of complicated external and internal flows has become a routine practice, replacing the expensive alternative of wind-tunnel testing. Successes that are mainly attributed to the rapid development of CFD include numerical modeling of the governing fluid physics, the ability to define the surfaces of a complicated geometry with volume-grid generation around these surfaces, and solution of the system of equations with efficient iterative solvers. Advanced research CFD codes such as CFL3D [1] and TLNS3D [2] are representative examples of the current state of the art in CFD.

The emerging field of CFD has reached a mature stage in which these codes can be employed in a multidisciplinary environment. In his review paper, Jameson [3] concluded that the following challenges remain to be met in the area of CFD: development of accurate higher order schemes; development of better schemes for capturing shocks and internal discontinuities; grid adaptation; use of unstructured grids to easily model the flow over and through complicated configurations; turbulence modeling; and design optimization.

The National Aeronautics and Space Administration (NASA) research efforts to incorporate high-fidelity single-discipline codes (including advanced CFD codes) in a multidisciplinary design procedure include the High-Speed Airframe Integration Research (HiSAIR) project [4] and the Computational Aerosciences (CAS) project of the High

Performance Computing and Communications (HPCC) program [5]. The HiSAIR project is primarily focused on High-Speed Civil Transport (HSCT) design activity, with the goal of developing advanced methodology and a computational environment for multidisciplinary analysis and design optimization. The HSCT is one application of the CAS project. These programs are committed to multidisciplinary design via a methodology known as sensitivity analysis (SA).

In reality, the interaction of many disciplines (including aerodynamics) must be considered in predicting the performance of an entire aircraft, and a methodology is needed to account for this interaction between the various disciplines. For example, the design of an aircraft wing involves the interaction of several disciplines (e.g., aerodynamics, structures, controls, and materials). Sobieski [6] (a pioneer in the development of the multidisciplinary approach) formulated a gradient-based multidisciplinary design (MdD) procedure based on the "divide and conquer" approach, where many disciplines are involved in the design process. This approach utilizes the required function response(s) of interest for each individual discipline, as well as the sensitivity derivatives (SD's) from each individual discipline (i.e., the derivatives of each individual discipline's output functions with respect to its input (design) variables). Sobieski [7] addressed the need to obtain SD's from advanced CFD codes, so that these codes can be used in a multidisciplinary design environment; furthermore, he derived the general individual-discipline discrete sensitivity equation, which is based on the implicit function theorem.

1.1 Literature Review

An SA is defined as the calculation of *slopes*, known as SD's, which are derivatives of the response(s) (output function(s)) of a particular system of interest taken with respect to the design variable(s) of interest. For the designer, an accurate knowledge of the SD's of a particular system under consideration can be used in many ways (e.g., for function approximation, trade-off design, and multidisciplinary design optimization (MDO)).

1.1.1 Sensitivity Analysis

Several procedures exist whereby the SD's can be obtained from advanced CFD codes. For example, these SD's can be calculated by using finite differencing, by hand differentiation, or by using symbolic manipulators, such as MACSYMA [8]. Alternatively, an automatic differentiation tool such as ADIFOR [9] can be used. A general yet conceptually simple method for computing aerodynamic SD's is the method of "brute force" finite differencing. For this method, under the assumption that forward finite-difference approximations are used, the CFD flow-analysis code is used to generate a single converged flow solution for a slightly perturbed value of each design variable for which SD's are required. Although this method of computing the SD's is used [10], there are several disadvantages:

1. Extremely high computational costs, particularly for three dimensions, because the number of flow analyses required in a typical design problem becomes large as the number of design variables becomes large.

2. Lack of robustness and accuracy because of difficulties that are sometimes associated with the selection of a proper numerical step size.

The step size can contribute to two types of errors in the finite-differencing method: approximation/truncation error and condition error. Truncation error is the difference between the exact value and the calculated value of the function. Condition error is due to computer round-off error that is associated with the subtraction of large numbers that are nearly equal. A trial-and-error approach is usually taken to determine a suitable step size when finite differencing is used; this approach can require many function evaluations. A method known as the finite-difference algorithm is outlined in Ref. [11] to automatically calculate an optimum step size. The finite-difference algorithm was extended in Ref. [12] to functions that are governed by matrix equations. This algorithm has not yet been demonstrated for cases in which the functions are calculated iteratively.

As an alternate approach that is typically less costly than finite differencing, aerodynamic SD's can (in principle) be computed by direct differentiation of the governing equations that control the fluid flow. Two approaches are commonly used: the discrete approach and the continuous approach. With the discrete approach, differentiation (with respect to the design variables) is of the discretized flow equations; with the continuous approach, differentiation is of the continuous governing equations using material derivatives or generalized calculus of variations. Differentiation via the continuous approach yields linear differential equations for the SD's; typically these differential sensitivity equations must be discretized and solved numerically for the required SD's. The discrete and continuous methods can yield identical SD's if the governing equations are self-adjoint (which is not the case for the Euler and Navier-Stokes equations) and if the discretization that is selected is the same for both methods; otherwise, the SD's obtained via the continuous method may not be consistent with the discrete function solutions. However, the advantage of using the continuous formulation is that of *flexibility*; (i.e., the governing equations and the discretization used for the SA can be different from that used for the flow analysis). An excellent review article by Taylor et al. [13] provides an overview of research activities in the efficient and accurate calculation of SD's with advanced CFD codes.

Early works by Pironneau [14] used the continuous formulation applied to the Navier-Stokes equations to derive sensitivity equations for incompressible low-Reynolds-number flow. Angrand [15] used a similar approach for flow over an airfoil using the irrotational flow (potential flow) approximation. Yates [16] and Yates and Desmarais [17] used a continuous formulation applied to the equations of linear aerodynamic theory and successfully obtained SD's from the integral-equation formulation of these governing equations in two dimensions. Extension of this method to three–dimensional (3-D) flow with the Navier-Stokes equations (for flow analysis and to calculate aerodynamic sensitivity derivatives) is possible, in principle. The integral-equation representation of

the governing equations has advantages over conventional finite-difference and finite-volume methods, and these advantages carry over to the solution of the resulting sensitivity equations.

Jameson [18, 19] and Jameson and Reuther [20] applied control theory to airfoil and wing design. They used a continuous formulation together with the adjoint-variable approach to obtain the required gradient information. Initially, their method was successfully implemented with conformal mapping for potential flow; more recently, they have extended it to inviscid flow in two and three dimensions with a finite-volume discretization. With this method, $2 + m$ flow analyses are required per design cycle, where two analyses are required to solve the flow equations and the adjoint equations (one analysis each) and m is the number of flow analyses required in the line-search procedure. The flow equations and the adjoint equations are solved efficiently by using the multigrid procedure in incremental iterative form.

Frank and Shubin [21], Shubin and Frank [22] and Shubin [23] obtained aerodynamic sensitivity equations using both the discrete and the continuous approaches. These studies indicates that consistent, discrete SD's should be used in aerodynamic design optimization; failure to do so can result in a considerable slowdown or complete failure of the optimization procedure. (Recall that the continuous method generally does not yield consistent, discrete SD's.)

With a continuous formulation, Borgaard and Burns [24] and Borgaard et al. [25] derived aerodynamic sensitivity equations in two dimensions by directly differentiating the Euler equations and the accompanying boundary conditions. Existing CFD software was easily modified to obtain the SD's with this approach. With this method, the nonlinear flow equations and linear flow-sensitivity equations were solved with the same solution procedure. However, in contrast to Frank and Shubin [21], Borgaard et al. concluded that judicious use of inconsistent, discrete SD's can sometimes result in successful optimization for cases in which the use of the consistent, discrete SD's sometimes fails.

With a continuous formulation, Ibrahim and Baysal [26] derived sensitivity equations in adjoint form and boundary (transversality) equations for the quasi-one-dimensional (quasi-1-D) Euler equations. This approach differs from other methods in that a perturbation technique is applied with a variation formulation to find the required gradient information. The resulting adjoint sensitivity equations and flow-analysis equations are solved with the same solution procedure because the character of these equations is similar. The method is applied to the optimization of a quasi-1-D nozzle, that includes a normal shock within the nozzle.

Elbanna and Carlson [27] applied the discrete sensitivity approach to calculate aerodynamic sensitivity coefficients in the transonic and supersonic flight regimes, where the governing equations of fluid flow considered are the transonic small-disturbance equations. Later, this approach is applied to the 3-D full-potential equation to compute aerodynamic sensitivity coefficients for a wing in a transonic flow. In order to avoid the excessive memory of a direct-solver approach, they used a conjugate-gradient iterative method to solve the very large system of linear sensitivity equations that is associated with 3-D flow. Elbanna and Carlson [28] used a symbolic manipulator, MACSYMA [8], to differentiate various parts of the 3-D full-potential flow code and successfully obtain these aerodynamic SD.

Baysal and Eleshaky [29], Baysal et al. [30], Burgreen et al. [31], and Eleshaky and Baysal [32] applied the discrete sensitivity approach to the steady Euler equations and later extended the approach to the thin-layer Navier-Stokes (TLNS) equations; results were presented for two-dimensional (2-D) flow. Taylor et al. [33, 34] and Hou et al. [35] also derived discrete sensitivity equations for the Euler and TLNS equations, with results given for 2-D flow. This discrete method results in very large systems of linear sensitivity equations that must be solved to obtain the SD's of interest. In Refs. 27 through 39, the sensitivity equations are solved in "standard" (i.e., nonincremental) form. Furthermore, in these references, a direct-solver method is applied to solve these

equations; the single exception is Ref. [39], where a hybrid direct/iterative approach is adopted for an isolated airfoil problem.

Eleshaky and Baysal [40] proposed a domain decomposition technique to solve the discrete sensitivity equations for large 2-D and 3-D problems. This method decomposes the large computational domain into subdomains; the sensitivity equations for the interior cells and the sensitivity equations for boundary cells that couple the subdomains are iteratively solved with a preconditioned conjugate gradient (CG) technique. The feasibility of computing the SD's on decomposed computational domains in two dimensions was demonstrated on a sample airfoil problem by Lacasse and Baysal [41]; in three dimensions it was demonstrated on an axisymmetric nacelle configuration by Eleshaky and Baysal [40].

Korivi et al. [42] and Newman et al. [43] proposed the incremental iterative method (IIM) to solve the sensitivity equation to calculate consistent, discrete SD's. With this approach, approximations of convenience can be introduced into the coefficient matrix operator without affecting the accuracy of the SD. The IIM enables the same solution strategy that is used to solve the equations of the flow analysis to be used to solve the flow sensitivity equations. This IIM strategy was first implemented in two dimensions for the TLNS equations with both the direct-differentiation and adjoint-variable approaches; the procedure was demonstrated for two airfoil problems: low-Reynolds-number laminar flow and high-Reynolds-number turbulent flow. In their work, the failure to differentiate the turbulence modeling terms (because of their complexity) resulted in inaccurate discrete SD's. Later, the IIM strategy was implemented in a 3-D marching Euler code to obtain SD's for several nongeometric design variables [44].

Chattopadhya and Pagaldipti [45] obtained quasi-analytical (discrete) SD's from the 3D parabolized Navier-Stokes equations and demonstrated the method for flow over a delta wing. In their study, grid sensitivity terms were first calculated via finite differences; in a later study [46], they were computed with a quasi-analytical method. Huddleston et

equations; the single exception is Ref. [39], where a hybrid direct/iterative approach is adopted for an isolated airfoil problem.

Eleshaky and Baysal [40] proposed a domain decomposition technique to solve the discrete sensitivity equations for large 2-D and 3-D problems. This method decomposes the large computational domain into subdomains; the sensitivity equations for the interior cells and the sensitivity equations for boundary cells that couple the subdomains are iteratively solved with a preconditioned conjugate gradient (CG) technique. The feasibility of computing the SD's on decomposed computational domains in two dimensions was demonstrated on a sample airfoil problem by Lacasse and Baysal [41]; in three dimensions it was demonstrated on an axisymmetric nacelle configuration by Eleshaky and Baysal [40].

Korivi et al. [42] and Newman et al. [43] proposed the incremental iterative method (IIM) to solve the sensitivity equation to calculate consistent, discrete SD's. With this approach, approximations of convenience can be introduced into the coefficient matrix operator without affecting the accuracy of the SD. The IIM enables the same solution strategy that is used to solve the equations of the flow analysis to be used to solve the flow sensitivity equations. This IIM strategy was first implemented in two dimensions for the TLNS equations with both the direct-differentiation and adjoint-variable approaches; the procedure was demonstrated for two airfoil problems: low-Reynolds-number laminar flow and high-Reynolds-number turbulent flow. In their work, the failure to differentiate the turbulence modeling terms (because of their complexity) resulted in inaccurate discrete SD's. Later, the IIM strategy was implemented in a 3-D marching Euler code to obtain SD's for several nongeometric design variables [44].

Chattopadhya and Pagaldipti [45] obtained quasi-analytical (discrete) SD's from the 3D parabolized Navier-Stokes equations and demonstrated the method for flow over a delta wing. In their study, grid sensitivity terms were first calculated via finite differences; in a later study [46], they were computed with a quasi-analytical method. Huddleston et

al. [47] applied the IIM strategy to calculate consistent, discrete SD's from a 2-D Euler-solver using the Gauss-Seidel algorithm with subiterations. The example used in their study was flow over an airfoil at subsonic and transonic flow conditions; they defined the shape of the airfoil with a Bezier-Bernstein parameterization. In their study, they note a discrepancy in the SD's when the quasi-analytical results are compared with the results obtained with finite differencing; this discrepancy is attributed to approximation of the derivatives of Roe's flux-difference-splitting scheme.

1.1.2 Design Optimization

Design optimization methods can be roughly classified as inverse design, gradient-based design, and nongradient-based design. Inverse aerodynamic design is a procedure in which typically a target surface-pressure distribution is specified, and the corresponding shape is calculated that will best produce this pressure profile. The disadvantage to this method is that physically realizable solutions may not exist. Thus, the inverse design problem must be carefully formulated. A review of inverse aerodynamic design methods is given in Ref. 48.

Nongradient-based optimization methods are based on genetic algorithms, simulated annealing techniques, and neural networks. Gradient-based techniques can be classified as either loosely coupled or tightly coupled optimizations. Loosely coupled optimization can also be called the "black box" method, in which the optimization software is implemented outside of the analysis cycles; the optimizer drives and controls the analysis and SA codes in the optimization procedure. The user typically can use the optimization code as a black box, in which the existing analysis and SA software are used for optimization without modifications. In the tightly coupled optimization procedure, the optimization cycles are embedded within (and are concurrent with) the iterations that are required in the function-analysis procedure. Gradient information is obtained concurrently within the procedure. The end result of the tightly coupled optimization procedure is the final improved design at convergence of the function-evaluation code. Gradient information for the loosely

coupled and tightly coupled methods can be obtained with either the discrete or the continuous approach.

Rizk [49] formulated a tightly coupled optimization procedure (also known as simultaneous analysis and design optimization) and summarized several CFD applications of this technique in Ref. 50. Ghattas and Xiaogang [51] used a discrete formulation to obtain the required gradient information and formulated a tightly coupled optimization procedure in an application to a low-Reynolds-number viscous flow. Hou et al. [52] successfully demonstrated tightly coupled optimization with a discrete adjoint formulation in application to a quasi-1-D nozzle problem. These two independent derivations of Hou and Ghattas arrive at essentially the same formulation for simultaneous aerodynamic analysis and design optimization; their methods are closely related to variational or control theory techniques. Ta'asan et al. [53] and Kuruvila et al. [54] used a continuous adjoint formulation to obtain gradient information and formulated the "one shot procedure," which is a tightly coupled optimization scheme in which a highly efficient multigrid method is used to solve the potential-flow equations and the accompanying adjoint sensitivity equation. With this method, the entire optimization procedure requires only about two to three times the computational cost of a single flow analysis. Huffman et al. [55] used a continuous adjoint formulation coupled with mesh sequencing to implement a simultaneous analysis and design optimization procedure in the TRANAIR code, which solves the full-potential equations of 3-D fluid flow. They employed a quasi-Newton-type solver to efficiently solve the flow analysis and adjoint sensitivity equations.

Other studies have recently been documented that present results for the loosely coupled aerodynamic optimization of wings using the 3-D Euler equations together with SD's calculated with either the discrete direct or discrete adjoint method. These studies were for transonic flow; therefore, they required a general 3-D flow solver (and appropriate computational grid) capable of solving mixed subsonic, transonic, and supersonic flows. For 3-D inviscid flow over a wing, Burgreen [56] and Burgreen and

Baysal [57, 58] considered both wing-section and planform design variables in their aerodynamic shape-optimization study. Jameson [59] considered wing-section variables only (for a fixed planform) and implemented an optimization technique based on control theory. Chattopadhya and Pagaldipti [45] developed a multidisciplinary, multilevel decomposition procedure for the optimal design of a high-speed transport wing with the parabolized Navier-Stokes equations and quasi-analytical aerodynamic SD.

Korivi et al. ([60] and the present study) use consistent, discrete SD's obtained by the direct-differentiation approach via the IIM with a space-marching algorithm for the Euler equations. Design-improvement studies are accomplished by using grid sensitivities from an automatically differentiated grid-generation code. The HSCT 24E configuration is chosen as the test case for the design-improvement studies in which only fully supersonic flow is considered.

1.2 Scope and Objective of the Present Study

The central focus of this study is to develop and demonstrate a methodology to efficiently calculate discrete (quasi-analytical) gradient information from advanced CFD codes. The IIM is proposed and successfully demonstrated in two dimensions to calculate these SD's. After successful demonstration in two dimensions, this methodology is extended to a 3-D marching Euler flow code to accurately and efficiently calculate geometric and non geometric SD's. Finally, a 3-D feasibility study (with the geometric SD) is done for the aerodynamic design improvement of the HSCT 24E configuration.

Fundamental sensitivity equations are derived by direct differentiation of the system of discrete nonlinear algebraic equations that model either the Euler or TLNS equations for 2-D and 3-D steady flows. This differentiation results in large systems of linear algebraic sensitivity equations that must be solved to obtain the derivatives of interest. Solving these sensitivity equations in standard form (i.e., nonincremental form) with a direct-solver approach is an option that has been investigated for some applications. Some important

advantages are realized in using a direct method when feasible. The lower/upper (LU) factorization of the coefficient matrix is stored in computer memory, and for multiple right-hand sides of the equation (corresponding to different design variables or different adjoint variables) the linear sensitivity equations can then be efficiently solved by the simple forward and backward substitution procedure. However, the most serious disadvantage of a direct method is the extremely large computer storage requirement, which appears to be well beyond the current capacity of modern supercomputers for practical 3-D problems; this capacity can even be exceeded in two dimensions on computational grids that contain a large number of points.

In an effort to circumvent the computer storage limitation for the direct methods, this study focuses on fundamental algorithm development for the efficient iterative solution of the aerodynamic sensitivity equations. The objective is to develop a solid framework in two dimensions from which extensions to three dimensions are proven feasible. In general, a serious difficulty encountered in the development and application of iterative techniques is the lack of diagonal dominance or poor overall conditioning in the coefficient matrix. Unfortunately, this problem is a very common occurrence in the CFD coefficient matrices of interest; the severity varies greatly and depends on many factors. This problem can manifest itself in either poor performance or even complete failure (i.e., divergence) of an iterative algorithm.

An "incremental" iterative method (also commonly known as the "delta" or "correction" form) is proposed in the present study to iteratively solve the aerodynamic sensitivity equations. This method has a computationally useful property that can be effectively exploited to combat the problems of poor iterative algorithm performance. This useful property allows the introduction of "approximations of convenience" into the coefficient-matrix operator of the equations without affecting the accuracy of the SD's at convergence. These approximations must be "reasonable" so that the resulting iterative strategy is convergent. In contrast, if approximations are made to the coefficient-matrix

operator of the equations in the standard form, then the computed SD cannot be consistent discrete forms; that is, they will not be the correct derivatives of the nonlinear algebraic equations that model the steady-state flow. In particular, it is proposed and successfully demonstrated numerically herein that the identical, diagonally dominant, approximate coefficient-matrix operator and algorithm, commonly associated with implicit methods for solving the nonlinear flow equations, can also be used to iteratively solve (in incremental form) the consistent, discrete systems of linear equations for aerodynamic SA.

The truly significant practical benefits of the proposed IIM can be realized only if the method can be successfully extended for use in three dimensions; this extension is demonstrated herein with the 3-D Euler equations. In particular, a space-marching algorithm together with the IIM is developed to calculate SD's in three dimensions; this method is applicable to fully supersonic, inviscid flow.

Another major part of this study focuses on the feasibility of applying the aerodynamic SD's to aerodynamic design optimization procedures in three dimensions; the HSCT 24E filleted-wing-body configuration (without nacelles and horizontal fins) is considered in this demonstration. A surface/volume-grid-generation code is differentiated to obtain the required grid-sensitivity terms, which are subsequently coupled with the SA code. The resulting SD's obtained via the IIM are compared on the basis of accuracy and efficiency with the same SD's obtained via finite differencing. The flow-analysis code, the differentiated surface/volume-grid-generation code, the aerodynamic SA code, and an optimizer code are coupled to make a complete aerodynamic design package. This design package is applied in three dimensions for thickness, camber, and planform design-improvement studies of the HSCT 24E configuration at supersonic cruise conditions.

The development of computer codes to conduct this study is summarized as follows. A 2-D Navier-Stokes computer code is developed with the capability to compute SD for geometric and nongeometric design variables via the IIM; this includes both the direct-differentiation and the adjoint-variable formulations. In particular, for accurate

and efficient applications to airfoil problems, the computer code is developed with a "lift-corrected" far-field boundary condition [61] for flow analysis and SA. A 3-D space-marching Euler code, MARSEN (marching Euler sensitivities), is developed for aerodynamic flow analysis, and the capability is developed for this code to compute SD's for geometric and nongeometric design variables using the IIM with the direct-differentiation approach.

1.3 Thesis Outline

This document is organized as follows. In Chap. 1, the introduction, literature review, and motivation have been presented. A brief review of the governing equations and method of solution is given in Chap. 2 for the 2-D Navier-Stokes equations, and necessary modifications are given for the space-marching algorithm applied to the Euler equations in three dimensions. The standard sensitivity equations with the direct differentiation and adjoint-variable approaches are given in Sec 3.1 and the IIM strategy is given in Sec. 3.2; the incremental iterative forms of these standard sensitivity equations are given in Sec. 3.3; a discussion with regard to the grid sensitivity is given in Sec. 3.4. The IIM methodology is extended to the space-marching Euler algorithm in three dimensions in Sec. 3.5. The SD's in two dimensions for a subsonic laminar case and a transonic turbulent case are given in Secs. 4.1 and 4.2, respectively. Similarly, the SD's in three dimensions for geometric and nongeometric design variables are given in Sec. 4.3. In Chap. 5, sample results are given from a feasibility study for design improvement of the HSCT 24E wing; SD's with respect to geometric design variables, coupled with an automatically-differentiated surface/volume-grid-generation code and an optimizer code are used. The summary, conclusions, and suggestions for further research are given in Chap. 6. The governing equations in curvilinear coordinates for the 2-D Navier-Stokes equations and for the 3-D Euler equations are given in Appendix A. The procedures for the linearization of a lift-corrected far-field boundary condition are given in Appendix B.

The adjoint-variable formulation in IIM form for inviscid flow with the space-marching algorithm is given in Appendix C. The parameterization of the HSCT 24E wing is given in Appendix D. Finally, a brief review of the automatic differentiation tool ADIFOR is given in Appendix E.

Chapter 2

GOVERNING EQUATIONS AND METHOD OF SOLUTION

In the present study, the governing equations for compressible, unsteady, inviscid flows in three dimensions and viscous flows in two dimensions are solved. These solutions are summarized in Appendix A. These equations are solved in the present study in their integral conservation-law form with a cell-centered finite-volume formulation [62, 63]. In this section, the procedure adopted to solve the 3-D Euler equations is outlined, and necessary modifications are suggested to handle the 2–D TLNS equations and 3-D space marching algorithm. The discretization of Eq. (A.1) in space and the application of the Euler implicit time discretization yields the following:

$$\left[\frac{1}{J\Delta t}\right]\{^n\Delta Q\} = \{R^{n+1}\} \tag{2.1}$$

Linearization of Eq. (2.1) about the nth time level yields

$$\left[\left[\frac{1}{J\Delta t}\right] - \left[\frac{\partial R^n}{\partial Q}\right]\right]\{^n\Delta Q\} = \{R^n(Q)\} \tag{2.2a}$$

$$\{^n\Delta Q\} = \{Q^{n+1}\} - \{Q^n\}$$

$$n = 1, 2, 3... \tag{2.2b}$$

In Eq. (2.2), $\left[\frac{1}{J\Delta t}\right]$ is a diagonal matrix, and $\left[\frac{\partial R^n}{\partial Q}\right]$ is a large, banded, sparse matrix. In this study, this Jacobian matrix plays another central role in the SA as discussed later. As the time step approaches infinity, Eq. (2.2) simply becomes the Newton-Raphson method for solving the nonlinear set of equations. Because we are interested only in

steady-state flow, the right-hand side of Eq. (2.2a) governs the physics of the fluid flow and the left-hand side is the matrix operator that governs the rate of convergence of the iterative procedure. The solution Q^* is the vector of field variables that corresponds to the residual at zero (i.e., the steady state). The residual $R(Q)$ includes the flux balances across each cell in the computational domain.

$$R(Q) = \delta \widehat{F}_\xi(Q) + \delta \widehat{G}_\eta(Q) + \delta \widehat{H}_\zeta(Q) \tag{2.3}$$

and

$$\left[\frac{\partial R}{\partial Q} \right] \triangle Q = \left[\delta \left(\frac{\partial \widehat{F}_\xi}{\partial Q} \triangle Q \right) + \delta \left(\frac{\partial \widehat{G}_\eta}{\partial Q} \triangle Q \right) + \delta \left(\frac{\partial \widehat{H}_\zeta}{\partial Q} \triangle Q \right) \right] \tag{2.4}$$

where $\widehat{F}_\xi, \widehat{G}_\eta$, and $\widehat{H}_\zeta$ are the inviscid flux terms in the ξ, η, and ζ curvilinear coordinate directions. The inviscid fluxes are calculated with the Van Leer upwind flux-vector-splitting method. Van Leer's flux vector splitting is chosen over other methods because with this method the fluxes are continuously differentiable at sonic and stagnation points; this feature is vital in the present study. Details of this method are given in Ref. 64. The terms $\delta \widehat{F}_\xi(Q)$ in Eq. (2.3) and $\delta \left[\frac{\partial \widehat{F}_\xi}{\partial Q} \triangle Q \right]$ in Eq. (2.4) are evaluated as

$$\delta \widehat{F}_\xi(Q) = \delta^- \widehat{F}_\xi^+(Q^-) + \delta^+ \widehat{F}_\xi^-(Q^+)$$

$$\delta \left[\frac{\partial \widehat{F}_\xi}{\partial Q} \triangle Q \right] = \delta^- \left(\frac{\partial \widehat{F}_\xi^+(Q^-)}{\partial Q} \triangle Q \right) + \delta^+ \left(\frac{\partial \widehat{F}_\xi^-(Q^+)}{\partial Q} \triangle Q \right) \tag{2.5}$$

where δ^- and δ^+ are backward and forward difference operators respectively. These fluxes are split into positive and negative parts based on the eigenvalues of the Jacobian matrices of the respective fluxes. Conserved variables Q are extrapolated from cell centers to cell faces in evaluating fluxes at cell interfaces based on the monotone upstream-centered schemes for conservative laws (MUSCL). The extrapolation procedure is accomplished with $\phi - \kappa$ interpolating polynomials given as

$$Q_{i+1/2}^- = Q_i + \frac{1}{4}\phi_\xi \left[(1 - \kappa_\xi) \nabla_\xi + (1 + \kappa_\xi) \triangle_\xi \right] Q_i$$

$$Q_{i+1/2}^+ = Q_{i+1} - \frac{1}{4}\phi_\xi \left[(1 - \kappa_\xi) \nabla_\xi + (1 + \kappa_\xi) \triangle_\xi \right] Q_{i+1} \tag{2.6}$$

where

$$\triangle_\xi Q_i = Q_{i+1} - Q_i, \quad \nabla_\xi Q_i = Q_i - Q_{i-1} \tag{2.7}$$

The value of ϕ determines whether extrapolation is first order ($\phi = 0$) or higher order ($\phi = 1$). Spatial accuracy is determined by the value of κ, where $\kappa = -1$ is second-order-accurate fully upwind, $\kappa = 1/3$ is third-order-accurate upwind biased (less than third accurate for multidimensional computations), and $\kappa = 1$ is equivalent to a second-order accurate central difference scheme. The subscript ξ denotes the direction in which the extrapolation is done. Similarly, expressions for $\widehat{G}_\eta$ and $\widehat{H}_\zeta$ are obtained by replacing i with j and k, respectively. With Eqs. (2.3) and (2.4), Eq. (2.2) can be written for a particular ijk^{th} interior cell as

$$\left[\frac{I}{J\triangle t} + B_\xi + B_\eta + B_\zeta \right] \triangle Q^n_{i,j,k}$$

$$+ D_\xi \triangle Q^n_{i-2,j,k} + A_\xi \triangle Q^n_{i-1,j,k} + C_\xi \triangle Q^n_{i+1,j,k} + E_\xi \triangle Q^n_{i+2,j,k}$$

$$+ D_\eta \triangle Q^n_{i,j-2,k} + A_\eta \triangle Q^n_{i,j-1,k} + C_\eta \triangle Q^n_{i,j+1,k} + E_\eta \triangle Q^n_{i,j+2,k}$$

$$+ D_\zeta \triangle Q^n_{i,j,k-2} + A_\zeta \triangle Q^n_{i,j,k-1} + C_\zeta \triangle Q^n_{i,j,k+1} + E_\zeta \triangle Q^n_{i,j,k+2}$$

$$= R_{i,j,k}(Q^n_{i,j,k}, Q^n_{i-2,j,k}, Q^n_{i-1,j,k}, Q^n_{i+1,j,k}, Q^n_{i+2,j,k}, Q^n_{i,j-2,k},$$

$$Q^n_{i,j-1,k}, Q^n_{i,j+1,k}, Q^n_{i,j+2,k}, Q^n_{i,j,k-2}, Q^n_{i,j,k-1}, Q^n_{i,j,k+1}, Q^n_{i,j,k+2}) \tag{2.8}$$

where $A_\xi, B_\xi, C_\xi, D_\xi$, and E_ξ are 5×5 block matrices in the ξ direction and similarly for the η and ζ directions. Equation (2.8) shows the left-hand side of the equation as a "thirteen point molecule" in a linear sense and the right-hand side of the equation represents the same molecule in a nonlinear sense. In two dimensions the block matrices are 4×4, and the block matrices $A_\eta, B_\eta, C_\eta, D_\eta$, and E_η are zero. Additional contributions to the block matrices and residual expression are made to account for the viscous terms, when applicable. The finite-volume equivalent of second-order-accurate central differences is used for the viscous terms. Details are given in Ref. 65. In two dimensions Eq. (2.8) can be written for a general ik^{th} interior cell as

$$\left[\frac{I}{J\Delta t} + B_\xi + B_\zeta\right]\Delta Q_{i,k}^n$$

$$+D_\xi\Delta Q_{i-2,k}^n + A_\xi\Delta Q_{i-1,k}^n + C_\xi\Delta Q_{i+1,k}^n + E_\xi\Delta Q_{i+2,k}^n$$

$$+D_\zeta\Delta Q_{i,k-2}^n + A_\zeta\Delta Q_{i,k-1}^n + C_\zeta\Delta Q_{i,k+1}^n + E_\zeta\Delta Q_{i,k+2}^n$$

$$= R_{i,k}(Q_{i,k}^n, Q_{i-2,k}^n, Q_{i-1,k}^n, Q_{i+2,k}^n, Q_{i+1,k}^n, Q_{i,k-2}^n, Q_{i,k-1}^n, Q_{i,k+1}^n, Q_{i,k+2}^n) \quad (2.9)$$

Adjustments have to be made to Eq. (2.8) in three dimensions and Eq. (2.9) in two dimensions near the boundaries. Furthermore, in the present study, all boundary condition relationships are consistently linearized (except lift-corrected far-field boundary conditions) and pre-eliminated in the global Jacobian matrix $\left[\frac{\partial R}{\partial Q}\right]$. References 39 and 65 provide more details regarding the linearization of boundary conditions. Inclusion of the linearization of the boundary conditions (discussed in Chap. 3) is of utmost importance in the present study. The structure of the global Jacobian matrix may change, depending on the type of boundary condition. For example, the implicit treatment of the periodic type of boundary condition results in off-diagonal terms inside or outside of the main bandwidth, depending on the ordering of the cells. Another example is the implicit treatment of the lift-corrected far-field boundary conditions [39], which couples the flow variables at the far field boundary with the flow variables on and adjacent to the surface boundary of the airfoil, and thus destroys the bandedness of the Jacobian matrix.

Equation (2.2) can be repeatedly solved with a direct solver (a Gaussian elimination solver) as the solution is advanced in time to steady state. Because of memory limitations, this method is not feasible for large 2-D and 3-D problems. The computational effort is reduced if first-order implicit discretization is used for the left-hand side of Eq. (2.2); this treatment does not affect the computational accuracy of the steady-state solution, which is determined by the spatial differencing of R(Q). Note that a first-order implicit discretization makes the left-hand-side Jacobian matrix of Eq. (2.2) block diagonally dominant and is represented by the approximate operator $\left[\widetilde{\frac{\partial R}{\partial Q}}\right]$. Typically, the differences

between the true Newton coefficient operator and the approximate coefficient-matrix operator include

(1) A "time-step" term is added, which significantly enhances each diagonal element of the coefficient matrix $\left[\frac{\widetilde{\partial R^n(Q)}}{\partial Q}\right]$. This addition is equivalent to the inclusion of underrelaxation in the true Newton's method and under certain restrictions can make the iterative procedure of Eq. (2.2) "time accurate".

(2) Simplifying linearization errors of various types are included in the construction of the approximate operator $\left[\frac{\widetilde{\partial R^n(Q)}}{\partial Q}\right]$. For example, consistent boundary-condition linearization is typically neglected, or a first-order accurate upwind treatment of the inviscid terms may be used in this matrix operator despite the higher order accurate treatment of these terms in the vector $R^n(Q)$ on the right-hand side of the equations.

(3) Additional "approximations of convenience" are included in the matrix operator in order that an efficient (in terms of computational work and computer storage) approximate solution of the linear problem can be generated at each iteration on the nonlinear problem. For example, with the popular, spatially split, approximate-factorization method of Ref. 66, an approximate solution of Eq. (2.2) is produced at each nth iteration with alternating direction sweeps that involve the solution of a series of uncoupled sub-systems of block-tridiagonal linear equations in each sweep direction. This algorithm is used in the sample problems for this study. Additional well-known iterative algorithms that have been applied to the solution of the Navier-Stokes equations include LU approximate factorization [67], conventional relaxation methods [68], strongly implicit methods [69], and preconditioned conjugate-gradient methods [70, 71].

In Eq. (2.8), $D_\xi, E_\xi, D_\eta, E_\eta, D_\zeta,$ and E_ζ are zero for the first-order implicit discretization. In three dimensions, supersonic flow is solved in a space-marching manner; this involves locally iterating in each crossflow plane, solving a local nonlinear problem,

before proceeding to the next cross plane. In fully supersonic flow, there is no upstream dependence on the downstream behavior. Equation (2.8) can be written for fully supersonic flow with first-order upwind discretization for the left-hand side as follows:

$$\left[\frac{I}{J\triangle t} + B_\xi + B_\eta + B_\zeta\right]\triangle Q_{i,j,k}^n$$

$$+ A_\eta\triangle Q_{i,j-1,k}^n + C_\eta\triangle Q_{i,j+1,k}^n + A_\zeta\triangle Q_{i,j,k-1}^n + C_\zeta\triangle Q_{i,j,k+1}^n$$

$$= R_{i,j,k}(Q_{i,j,k}^n, Q_{i-2,j,k}^*, Q_{i-1,j,k}^*, Q_{i,j-2,k}^n,$$

$$Q_{i,j-1,k}^n, Q_{i,j+1,k}^n, Q_{i,j+2,k}^n, Q_{i,j,k-2}^n, Q_{i,j,k-1}^n, Q_{i,j,k+1}^n, Q_{i,j,k+2}^n) \qquad (2.10)$$

In Eq. (2.10), the coefficient of $\triangle Q_{i+1}$, C_ξ, is zero for fully supersonic flow. Space marching is done in the direction of the flow (i.e., the i direction in the present study). Information in the previous cross plane is known when iterating locally in the ith cross-flow plane [72] (i.e., Q_{i-1}^* and Q_{i-2}^* are the steady-state flow variables in the i-1 and i-2 cross planes respectively). For this reason, the term $\triangle Q_{i-1,j,k}$ is zero and not included in Equation (2.10) for the present space-marching algorithm. Equation (2.10) can be expressed as

$$\left[M + B_\eta + B_\zeta\right]\triangle Q_{i,j,k}^n$$

$$+ A_\eta\triangle Q_{i,j-1,k}^n + C_\eta\triangle Q_{i,j+1,k}^n + A_\zeta\triangle Q_{i,j,k-1}^n + C_\zeta\triangle Q_{i,j,k+1}^n$$

$$= R_{i,j,k}(Q_{i,j,k}^n, Q_{i-2,j,k}^*, Q_{i-1,j,k}^*, Q_{i,j-2,k}^n,$$

$$Q_{i,j-1,k}^n, Q_{i,j+1,k}^n, Q_{i,j+2,k}^n, Q_{i,j,k-2}^n, Q_{i,j,k-1}^n, Q_{i,j,k+1}^n, Q_{i,j,k+2}^n) \qquad (2.11)$$

where

$$M = \left[\frac{I}{J\triangle t} + B_\xi\right]$$

Equation (2.11) is approximately factored as

$$\left[(M + B_\eta)\triangle Q_{i,j,k}^n + A_\eta \triangle Q_{i,j-1,k}^n + C_\eta \triangle Q_{i,j+1,k}^n \right] \left[M^{-1} \right]$$

$$\left[(M + B_\zeta)\triangle Q_{i,j,k}^n + A_\zeta \triangle Q_{i,j,k-1}^n + C_\zeta \triangle Q_{i,j,k+1}^n \right]$$

$$= R_{i,j,k}(Q_{i,j,k}^n, Q_{i-2,j,k}^*, Q_{i-1,j,k}^*, Q_{i,j-2,k}^n,$$

$$Q_{i,j-1,k}^n, Q_{i,j+1,k}^n, Q_{i,j+2,k}^n, Q_{i,j,k-2}^n, Q_{i,j,k-1}^n, Q_{i,j,k+1}^n, Q_{i,j,k+2}^n) \qquad (2.12)$$

The solution of Eq. (2.12) involves the solution of two block-tridiagonal equations. The preceding equation can be written compactly for the ith crossplane as

$$\left[(M + B_\eta), A_\eta, C_\eta \right]_i \Phi_i = R_i^n(Q_i^n)$$

$$\left[(M + B_\zeta), A_\zeta, C_\zeta \right]_i {}^n\triangle Q_i = M_i \Phi_i$$

$${}^n\triangle Q_i = Q_i^{n+1} - Q_i^n \quad n = 1, 2, 3 \qquad (2.13)$$

where Φ_i is the intermediate solution for the ith crossplane. The flow variables are solved and updated at each iteration as shown in Eq. (2.13).

Chapter 3

DISCRETE SENSITIVITY ANALYSIS

In Sec. 3.1 of this chapter, fundamental sensitivity equations are derived for two dimensions in standard form with the direct-differentiation and adjoint-variable approaches. In Sec. 3.2, the incremental method for solving the linear system of equations is discussed. Later, in Sec. 3.3, the standard sensitivity equations are cast in incremental iterative form in two dimensions. Various methods for calculating mesh sensitivity are discussed in Sec. 3.4. In Sec. 3.5, the IIM is extended to solve the sensitivity equations in three dimensions with a space-marching procedure for supersonic Euler flow with the direct-differentiation approach.

3.1 Fundamental Aerodynamic Sensitivity Equations in Standard Form

In general, the jth aerodynamic system response C_j is functionally dependent on the vector of steady-state field variables $\{Q^*\}$, the vector of the computational grid (x,y) coordinates, $\{\bar{X}\}$, and perhaps also explicitly on the vector of independent design variables $\bar{\beta}$. That is,

$$C_j = C_j(Q^*(\bar{\beta}), \bar{X}(\bar{\beta}), \bar{\beta}) \tag{3.1}$$

The SD of C_j with respect to the kth design variable β_k (i.e., the kth element of $\bar{\beta}$) is, thus,

$$\frac{dC_j}{d\beta_k} = \left\{\frac{\partial C_j}{\partial Q}\right\}^T \left\{\frac{dQ^*}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_j}{\partial \beta_k} \tag{3.2}$$

where the superscript T denotes transpose.

The notation for a total derivative has been used on the left-hand side of Eq. (3.2) which indicates that the total rate of change of C_j with respect to β_k is included in the

term and distinguishes it from the partial derivative on the right-hand side of the equation. Nevertheless, $\frac{dC_j}{d\beta_k}$ is a partial derivative in the sense that C_j is generally a function of multiple independent design variables $\bar{\beta}$, as seen in Eq. (3.1). In Eq. (3.2), the term $\left\{\frac{d\bar{X}}{d\beta_k}\right\}$ is known as the grid-sensitivity vector; a detailed discussion is given in Sec. 3.4. The grid-sensitivity vector is null if the design variable β_k is not related to the geometric shape of the domain. The vector $\left\{\frac{dQ^*}{d\beta_k}\right\}$, which is the sensitivity of the steady-state field variables with respect to the kth design variable, is evaluated for use in Eq. (3.2) by solving a large system of coupled linear sensitivity equations.

The large system of coupled nonlinear algebraic residual equations that model the fluid flow can be generally expressed as

$$\{R(Q^*(\bar{\beta}),\bar{X}(\bar{\beta}),\bar{\beta},C_L)\} = \{0\} \tag{3.3}$$

where the dependence of these equations on the grid $\{\bar{X}\}$ and on the design variables $\bar{\beta}$ is noted. In addition, Eq. (3.3) includes the possibility of an explicit dependence on the steady-state lift coefficient C_L. This explicit dependence is found in the far-field boundary conditions of an isolated lifting airfoil when the accurate, "lift-corrected" far-field boundary conditions of Ref. [61] have been used, as in the 2-D sample problems of this study. Note that C_L itself depends on the field variables $\{Q^*\}$, the grid $\{\bar{X}\}$, and possibly explicitly on the design variables $\bar{\beta}$, in the manner expressed by Eq. (3.1). The explicit dependence on C_L noted in Eq. (3.3) might, therefore, appear redundant; however, the computational advantages of this particular grouping of terms is discussed in detail in Ref. [39] and will become apparent subsequently.

Differentiation of Eq. (3.3) with respect to β_k yields

$$\left\{\frac{dR}{d\beta_k}\right\} = \left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ^*}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R}{\partial \beta_k}\right\} + \left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k} = \{0\} \tag{3.4}$$

where in Eq. (3.4) the term $\frac{dC_L}{d\beta_k}$ is evaluated with a relationship of the form given by Eq. (3.2). Note that the vector $\left\{\frac{\partial R}{\partial C_L}\right\}$ is very sparse; nonzero contributions to it arise only from the "lift-corrected" far-field boundary-condition equations. Equation (3.4) is,

thus, a large system of coupled linear equations that can in principle be solved for the unknown vector $\left\{ \frac{dQ^*}{d\beta_k} \right\}$; one such solution is obtained for each design variable β_k. This method is known as the quasi-analytical method for computing SD's.

The matrix $\left[\frac{\partial R}{\partial Q} \right]$ of Eq. (3.4) is the Jacobian of the nonlinear flow equations (evaluated at steady state) with respect to the field variables and includes consistent treatment of all boundary conditions; an exception is the contribution that results from the explicit dependence of the lift-corrected far-field boundary conditions on C_L. Substitution of Eq. (3.2) for $\frac{dC_L}{d\beta_k}$ into Eq. (3.4) reveals that this contribution to $\left[\frac{\partial R}{\partial Q} \right]$ is given by the very sparse matrix $\left\{ \frac{\partial R}{\partial C_L} \right\} \left\{ \frac{\partial C_L}{\partial Q} \right\}^T$. The matrix $\left[\frac{\partial R}{\partial X} \right]$ of Eq. (3.4) is the Jacobian of the flow equations (evaluated at the steady state and including all boundary conditions) with respect to the grid coordinates [33–37]; again, the exception is the contribution from the explicit dependence of the far-field boundary conditions on C_L. Here, this contribution is given by the very sparse matrix $\left\{ \frac{\partial R}{\partial C_L} \right\} \left\{ \frac{\partial C_L}{\partial X} \right\}^T$. The vector $\left\{ \frac{\partial R}{\partial \beta_k} \right\}$ of Eq. (3.4) accounts for explicit dependencies (if any) of the flow equations (including boundary conditions) on β_k; the contribution to this vector from the C_L dependence of the far-field boundary conditions is given by the vector $\left\{ \frac{\partial R}{\partial C_L} \right\} \frac{\partial C_L}{\partial \beta_k}$. More details in regard to the inclusion of lift-corrected far-field boundary conditions are given in Appendix B.

The Jacobian matrix $\left[\frac{\partial R}{\partial Q} \right]$ must include consistent linearization of boundary conditions. This inclusion can be done with or without pre-elimination, the details of which are given in Ref. [35]. With pre-elimination, one expresses the boundary unknowns in terms of the interior unknowns, whereas without pre-elimination one solves the interior and boundary unknowns simultaneously. Inclusion of the linearization of boundary conditions in the Jacobian matrix is very important to obtaining accurate SD's as noted by Hou et al. [35].

A well-known, closely related alternate strategy for computing SD's known as the adjoint -variable method, is easily developed with expressions that have been presented

thus far. The development begins by combining Eqs. (3.2) and (3.4) to yield

$$\frac{dC_j}{d\beta_k} = \left\{\frac{\partial C_j}{\partial Q}\right\}^T \left\{\frac{dQ^*}{d\beta_k}\right\} + \left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_j}{\partial \beta_k}$$
$$+ \{\lambda_j\}^T \left(\left[\frac{\partial R}{\partial Q}\right]\left\{\frac{dQ^*}{d\beta_k}\right\} + \left[\frac{\partial R}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R}{\partial \beta_k}\right\} + \left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k}\right) \qquad (3.5)$$

The adjoint-variable vector $\{\lambda_j\}$ is arbitrary at this point because the inner product of $\{\lambda_j\}$ is taken with the null vector, from Eq. (3.4). Thus, no net change occurs from Eq. (3.2) to Eq. (3.5) because the entire additional term on the right-hand side of Eq. (3.5) is zero for any and all $\{\lambda_j\}$. Expansion and rearrangement of Eq. (3.5) yields

$$\frac{dC_j}{d\beta_k} = \left(\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial \bar{X}}\right]\right)\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_j}{\partial \beta_k} + \{\lambda_j\}^T\left\{\frac{\partial R}{\partial \beta_k}\right\}$$
$$+ \{\lambda_j\}^T\left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k} + \left(\left\{\frac{\partial C_j}{\partial Q}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial Q}\right]\right)\left\{\frac{dQ^*}{d\beta_k}\right\} \qquad (3.6)$$

The necessity of evaluating the vector $\left\{\frac{dQ^*}{d\beta_k}\right\}$ with Eq. (3.4) is eliminated for all β_k by selecting the vector $\{\lambda_j\}$ such that the coefficient of $\left\{\frac{dQ^*}{d\beta_k}\right\}$ in Eq. (3.6) is null. That is, select $\{\lambda_j\}$ so that it satisfies

$$\left\{\frac{\partial C_j}{\partial Q}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial Q}\right] = \{0\}^T \qquad (3.7)$$

or

$$\left[\frac{\partial R}{\partial Q}\right]^T\{\lambda_j\} + \left\{\frac{\partial C_j}{\partial Q}\right\} = \{0\} \qquad (3.8)$$

Therefore, Eq. (3.8) is solved for this particular choice of the adjoint-variable vector $\{\lambda_j\}$, the SD's of C_j with respect to all β_k are computed by

$$\frac{dC_j}{d\beta_k} = \left(\left\{\frac{\partial C_j}{\partial \bar{X}}\right\}^T + \{\lambda_j\}^T\left[\frac{\partial R}{\partial \bar{X}}\right]\right)\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C_j}{\partial \beta_k} +$$
$$\{\lambda_j\}^T\left\{\frac{\partial R}{\partial \beta_k}\right\} + \{\lambda_j\}^T\left\{\frac{\partial R}{\partial C_L}\right\}\frac{dC_L}{d\beta_k} \qquad (3.9)$$

Note that Eq. (3.9) can be solved for $\frac{dC_j}{d\beta_k}$ only if $\frac{dC_L}{d\beta_k}$ is known or if $C_j = C_L$. Therefore, when the lift-corrected far-field boundary conditions are treated in the manner described, then $\frac{dC_L}{d\beta_k}$ must be the first SD that is calculated (for any and all β_k of concern), regardless of whether the sensitivity of C_L is of actual interest. (Typically, of course, the SD's of C_L will be of interest in most problems.) A particular solution $\{\lambda_j\}$ is valid only for a specific system response C_j; thus, the solution of Eq. (3.8) must be repeated for each different system response of interest.

We can easily verify from the preceding equations that each solution $\left\{\frac{dQ^*}{d\beta_k}\right\}$ of Eq. (3.4) for a particular design variable can be used for an unlimited number of different system responses. In contrast, however, each solution $\{\lambda_j\}$ of Eq. (3.8) for a particular system response can be used for an unlimited number of different design variables. Therefore, the total number of large linear systems that must be solved for a particular problem can be minimized through a judicious selection of one of these two methods, depending on whether the number of system responses of interest or the number of design variables of interest is larger.

In terms of computational efficiency, the significance of the difference in the two methods is diminished greatly if a direct method is used to solve these linear systems (i.e., either Eq. (3.4) or (3.8)). The difference is diminished because with either method the LU factorization must only be done once and is then repeatedly reused for multiple right-hand-side vectors. However, this distinction can become very important if an iterative strategy is used to solve these linear systems, particularly if the difference between the number of design variables and the number of system responses of interest is very large. Despite this difference, these two methods are equivalent in the sense that they yield identical values for the SD, if properly implemented computationally.

To briefly summarize, the calculation of the aerodynamic SD's with both the discrete direct differentiation and adjoint methods requires the direct or iterative solution of large linear systems of equations of the type given by either Eq. (3.4) or (3.8). These two

systems of linear equations are referred to as the "aerodynamic sensitivity equations in standard form." Fundamental algorithm development for the solution of one of these two linear systems is easily extended and applied to the other because their respective coefficient matrices $\left[\frac{\partial R}{\partial Q}\right]$ and $\left[\frac{\partial R}{\partial Q}\right]^T$ are transposes of each other. When the standard-form equations are solved, no approximations can be introduced into any of the terms without simultaneously introducing error into the resulting SD's. In this form, the framework to support the development of iterative methods is thus rigid and restrictive.

As a consequence, given the choice of a higher order accurate upwind approximation for the spatial discretization of the flow analysis, a consistent, higher order accurate, upwind spatial discretization, including a fully consistent treatment of all boundary conditions, is required in the coefficient-matrix operator of the sensitivity equations (in standard form). Furthermore, no "time term" can be added here to enhance each element of the diagonal, as is used (in contrast) in the implicit formulation for solving the nonlinear flow equations. Unfortunately, the resulting coefficient matrix (either $\left[\frac{\partial R}{\partial Q}\right]$ or $\left[\frac{\partial R}{\partial Q}\right]^T$) of the linear sensitivity equations in standard form in this case is not block-diagonally dominant [68]; consequently, the computational performance of traditional iterative methods for solving these equations in this standard form is expected to be poor or even to fail [39]. Therefore, this particular difficulty (i.e., the lack of sufficient diagonal dominance) and its resolution are of principal concern in the development of the incremental form of the equations in the following sections.

3.2 Basic Linear Equation Solution in Incremental Form

Consider the linear system of algebraic equations in the general form

$$[A]\{Z^*\} + \{B\} = \{0\} \tag{3.10}$$

where $\{Z^*\}$ is the solution vector. In treating the problem of solving Eq. (3.10), which is essentially a "root finding" problem, the application of Newton's method (traditionally

used in root finding for nonlinear equations) to the linear problem yields the basic two-step incremental iterative formulation

$$-[\widetilde{A}]\{{}^{m}\triangle Z\} = [A]\{Z^{m}\} + \{B\} \tag{3.11}$$

$$\left\{Z^{m+1}\right\} = \{Z^{m}\} + \{{}^{m}\triangle Z\}$$

$$m = 1, 2, 3, ... \tag{3.12}$$

where m is an iteration index and $\{{}^{m}\triangle Z\}$ is the incremental change in the solution from the known (m^{th}) to the next $(m^{th}+1)$ iteration level. An initial guess $\left\{Z^{1}\right\}$ is required to begin the procedure, which in the present study is taken everywhere as zero. If Newton's method is applied strictly, the coefficient matrix $[\widetilde{A}]$ is equal to the matrix $[A]$, and clearly the two-step iterative strategy of Eqs. (3.11) and (3.12) for the linear problem converges on the first iteration for any initial guess. Therefore, in this case the solution of the linear system in the standard form (Eq. (3.10)) and the solution in the incremental form (Eqs. (3.11) and (3.12)) are equivalent.

More generally, however, the matrix $[\widetilde{A}]$ is not necessarily equal to the matrix $[A]$. The matrix $[\widetilde{A}]$ can be any convenient approximation of the matrix $[A]$ with the restriction that $[\widetilde{A}]$ must approximate $[A]$ well enough so that the two-step iterative procedure (Eqs. (3.11) and (3.12)) converges (or at the very least can be forced to converge by including a strategy such as underrelaxation). Simply stated, $[\widetilde{A}]$ should capture the essence of $[A]$. Furthermore, because the equations have been cast in delta form, the incremental method produces the unique solution of Eq. (3.10), $\{Z^{*}\}$, if convergent. In this formulation, the purpose of the left-hand-side operator is to drive the right-hand-side vector to zero; the accuracy of the unknown $\{Z^{*}\}$ depends on the right-hand side and any approximations to the right-hand side result in erroneous final results.

Equation (3.11) can be solved with either a direct solver or an iterative solver. With the direct solver, the left-hand-side operator of Eq. (3.11) is LU factorized and

stored. This LU factored matrix is reused for multiple right-hand sides with forward and backward substitutions for multiple iterations. For large problems in two and three dimensions, iterative algorithms are the only choice because of the restrictions on computer memory. If an iterative algorithm with inner iterations is introduced for solving Eq. (3.11) then the the iteration cycle over Eqs. (3.11) and (3.12) becomes the outer iteration index. The inner iterative procedure convergence is ensured if the left-hand-side matrix approximation is block-diagonally dominant. The outer iterative procedure convergence is ensured, as discussed previously, if the approximate operator is an adequate approximation to the matrix [A] and, when inner iterations are included, if the inner iterative procedure is converged to some satisfactory tolerance (whatever that tolerance may be).

For example, for selection of a conventional relaxation algorithm to solve Eq. (3.11), the matrix $-\widetilde{[A]}$ is split into two parts as

$$-\left[\widetilde{A}\right] = [M] + [N] \tag{3.13}$$

The IIM becomes

$$\text{Step 1}: \quad [M]\left\{{}^{m,i}\Delta Z\right\} = [A]\{Z^m\} + \{B\} - [N]\left\{{}^{m,i-1}\Delta Z\right\}S$$
$$i = 1, 2, 3, ...(\text{imax})^m$$

$$\text{Step 2}: \quad \left\{Z^{m+1}\right\} = \{Z^m\} + \left\{{}^{m,(\text{imax})^m}\Delta Z\right\}$$
$$m = 1, 2, 3, ... \tag{3.14}$$

where $(imax)^m$ is the number of inner or subiterations to converge the mth linear subproblem at step 1 to some desired tolerance. The splitting of the matrix as in Eq. (3.13) is chosen such that Eq. (3.14) can be repeatedly solved efficiently in terms of CPU time and memory requirement. Popular choices for splitting the matrix yield either the Jacobi or the Gauss-Siedel algorithms of either the point or line-relaxation types. More details

are given in Ref. [65] in which the delta-form line Gauss-Siedel algorithm with inner and outer iterations is chosen to solve the nonlinear 2–D fluid equations.

Advantages of using the IIM can be summarized as follows:

(1) Iterative algorithms can be used to solve the sensitivity equations in incremental iterative form efficiently. In contrast, for solution of the standard form of these equations, iterative algorithms may converge very slowly or even may result in complete failure; this is because of the lack of block-diagonal dominance in the higher order Jacobian matrix.

(2) The same approximate operator available for solving the flow equations in most implicit CFD codes can also be used to solve the sensitivity equation; thus a time term that acts as an under relaxation parameter can be added to the approximate operator in incremental iterative form.

(3) Solution of the sensitivity equation via the IIM requires less computer memory than solution of the sensitivity equation in standard form with in-core banded solvers. This reduction in memory enables solution of large 2-D and 3-D problems.

(4) Tools like ADIFOR can be used to compute the right-hand side of the sensitivity equation efficiently and accurately even when complicated turbulence models are being used.

3.3 Incremental Solution of the Equations of Aerodynamic Sensitivity Analysis

Application of the fundamental incremental formulation for solution of the linear equation (Eqs. (3.11) and (3.12)) to the linear system of Eq. (3.4) (i.e., the quasi-analytical method) for computing aerodynamic SD's gives

$$-\left[\widetilde{\frac{\partial R}{\partial Q}}\right]\left\{{}^{m}\triangle\frac{dQ}{d\beta_k}\right\} = \left\{\frac{dR^m}{d\beta_k}\right\} \tag{3.15}$$

$$\left\{\frac{dQ^{m+1}}{d\beta_k}\right\} = \left\{\frac{dQ^m}{d\beta_k}\right\} + \left\{{}^{m}\triangle\frac{dQ}{d\beta_k}\right\}$$

$$m = 1, 2, 3, \ldots \ldots \tag{3.16}$$

where

$$
\left\{ \frac{dR^m}{d\beta_k} \right\} = \left[\frac{\partial R}{\partial Q} \right] \left\{ \frac{dQ^m}{d\beta_k} \right\} + \left[\frac{\partial R}{\partial \bar{X}} \right] \left\{ \frac{d\bar{X}}{d\beta_k} \right\} + \left\{ \frac{\partial R}{\partial \beta_k} \right\} + \left\{ \frac{\partial R}{\partial C_L} \right\} \frac{dC_L^m}{d\beta_k}
$$

$$
\frac{dC_L^m}{d\beta_k} = \left\{ \frac{\partial C_L}{\partial Q} \right\}^T \left\{ \frac{dQ^m}{d\beta_k} \right\} + \left\{ \frac{\partial C_L}{\partial \bar{X}} \right\}^T \left\{ \frac{d\bar{X}}{d\beta_k} \right\} + \frac{\partial C_L}{\partial \beta_k} \tag{3.17}
$$

where the left-hand-side coefficient-matrix operator $\left[\widetilde{\frac{\partial R}{\partial Q}} \right]$ approximates the matrix $\left[\frac{\partial R}{\partial Q} \right]$ (which will be discussed subsequently). The vector $\left\{ \frac{dR^m}{d\beta_k} \right\}$ represents the mth iteration on the total derivative of the discrete steady-state nonlinear flow equations (Eq. (3.3)), with respect to β_k. From Eq. (3.4), clearly this vector must be driven to zero to find the solution $\left\{ \frac{dQ^*}{d\beta_k} \right\}$ of Eq. (3.4), which, is of course, the objective of the incremental strategy of Eqs. (3.15), (3.16), and (3.17). Approximations must not be made to any terms of the vector $\left\{ \frac{dR^m}{d\beta_k} \right\}$; in particular, a consistent treatment of all boundary conditions is necessary if the converged solution is to yield the correct, consistent, discrete SD's. The final solution at convergence depends only on the terms of this right-hand-side vector.

The identical approximate left-hand-side coefficient-matrix operator $\left[\widetilde{\frac{\partial R}{\partial Q}} \right]$ and algorithm, which are used to solve the nonlinear problem for the flow variables, are also proposed for use (when evaluated at the steady state) as the approximate left-hand-side operator and algorithm that are used in solving the linear equation (Eq. (3.15)) for the flow sensitivities. That is, a first-order-accurate upwind spatial discretization of the inviscid terms is used in this operator as an approximation here to the higher order accurate, upwind discretization of these terms. Note that as a result of this choice, block-diagonal dominance is obtained and maintained in the left-hand side coefficient matrix. In addition, a false "time term" is included (i.e., added) so that each diagonal element

of the matrix $\left[\widetilde{\frac{\partial R}{\partial Q}}\right]$ is further enhanced; this additional term is equivalent to under-relaxation in the incremental strategy shown in Eqs. (3.15), (3.16), and (3.17). The boundary conditions are not linearized in a fully consistent manner in this approximate matrix operator; far off-diagonal contributions from the periodic boundary conditions which arise when calculations are performed on a C- or O-mesh are neglected. However, these periodic boundary conditions cause computational difficulties for the standard-form equations which require a consistent treatment in the left-hand-side matrix operator [38]. Finally, the well-known spatially split approximate factorization algorithm [66] (also used here to solve the nonlinear flow equations) is used to solve Eq. (3.15) (approximately) at each mth iteration. If the resulting block-tridiagonal coefficient matrices are stored over the entire domain, only a single LU factorization of each coefficient matrix is required. Hence, the coefficient matrix is reused for all iterations and all design variables. This strategy is implemented in the large 2–D sample problems presented.

If the adjoint-variable formulation for computing the SD is preferred, then application of the incremental formulation for solution of the linear equation (Eqs. (3.11) and (3.12)) to the linear system of Eq. (3.8) for computing the adjoint-variable vector $\{\lambda_j\}$ yields

$$-\left[\widetilde{\frac{\partial R}{\partial Q}}\right]^T \{{}^m\triangle\lambda_j\} = \left[\frac{\partial R}{\partial Q}\right]^T \{\lambda_j^m\} + \left\{\frac{\partial C_j}{\partial Q}\right\} \tag{3.18}$$

$$\left\{\lambda_j^{m+1}\right\} = \left\{\lambda_j^m\right\} + \{{}^m\triangle\lambda_j\}$$
$$m = 1, 2, 3, \ldots \ldots \tag{3.19}$$

For application in Eq. (3.18), the approximate left-hand-side coefficient-matrix operator and algorithm (described previously for use in Eq. (3.15)) can be easily transposed. Again, only a single LU factorization of the globally stored block-tridiagonal coefficient matrices is required.

3.4 Grid (Mesh) Sensitivity

In this section, the sensitivity of the grid or mesh with respect to the design variables is discussed. The computational grids used in CFD usually are body-fitted grids. Movement of the boundary because of changes in the design variables affects the entire computational grid. This term is not zero, and, thus, it needs special consideration.

One method for computing this quantity $\left\{ \frac{d\bar{X}}{d\beta} \right\}$ is to use divided differences. Each design variable is perturbed, and a new mesh is generated; mesh sensitivity is calculated from

$$\left\{ \frac{d\bar{X}}{d\beta_k} \right\} = \frac{\bar{X}(\beta_k + \triangle\beta_k) - \bar{X}(\beta_k - \triangle\beta_k)}{2\triangle\beta_k} \tag{3.20}$$

where central differences are used and $\triangle\beta_k$ is the change in the kth component of the design-variable vector $\bar{\beta}$. This method can be used only for those grid-generation techniques that provide the same number of cells when the design variable is perturbed as in the original mesh. Grid-generation equations by formulation are smooth compared with the governing equations of fluid flow; finite differencing can provide a good approximation. The disadvantage to using this method is its computational cost. If hyperbolic or elliptic grid-generation techniques are adopted, this method for computing grid sensitivity becomes expensive, particularly when these grid-generation tools are used in an automated design environment. Moreover, sophisticated grid-generation tools are interactive, which prohibits their use in an automated design loop.

One method for calculating grid sensitivity is to make use of an automatic-differentiation (AD) tool to obtain grid sensitivity. Green et al. [77] applied the automatic-differentiation tool ADIFOR to obtain the grid sensitivity from a 3-D algebraic grid generator and successfully obtained SD's from an AD -enhanced version of the TLNS3D flow code for turbulent flow over an ONERA M6 wing. In the present study, grid sensitivity in three dimensions is obtained from an automatic surface/volume-grid-generator code [80] by using the AD tool, and the resultant grid sensitivity is successfully

used in a gradient-based design improvement of the HSCT 24E configuration. This method can be expensive if iterative grid-generation techniques are used.

Alternatively, a method of avoiding the evaluation of grid sensitivity and expensive regridding in a design loop is the use of using transpiration [22]. With this method, one can approximately compute $\left[\frac{\partial R}{\partial X}\right]\left\{\frac{dX}{d\beta_k}\right\}$ and avoid grid generation when the geometry shape changes. The zero flux through the boundary is modified on the surface to a fixed value to approximate what would have happened if the body shape had actually changed. However, this method requires considerable care to compute accurate SD's and model real surface mass transpiration in Navier-Stokes simulations.

A computationally efficient technique is proposed in Ref. [34] that involves the chain rule and analytical differentiation of the relationships used to distribute the mesh points in the computational domain. Boundary coordinates $\bar{X}_s$ can be viewed as principal input to the grid coordinates in the rest of the domain, and these boundary coordinates are defined by some parametric relationship that involves the design variables. Thus, the grid generation procedure can be represented as

$$\bar{X} = \bar{X}(\bar{X}_s(\bar{\beta}))$$ (3.21)

The grid-sensitivity term obtained by differentiating Eq. (3.21) with respect to the design variable $\bar{\beta}_k$ is

$$\left\{\frac{d\bar{X}}{d\beta_k}\right\} = \left[\frac{\partial\bar{X}}{\partial\bar{X}_s}\right]\left\{\frac{\partial\bar{X}_s}{\partial\beta_k}\right\}$$ (3.22)

where the matrix $\left[\frac{\partial\bar{X}}{\partial\bar{X}_s}\right]$ in Eq. (3.22) is unique to a particular grid-generation program and needs to be constructed only once. Smith and Sadrehaghighi [73] and Sadrehaghighi et al. [74] applied this approach and obtained the grid sensitivity for a 2-D algebraic grid generator TBGG (twin-boundary grid generation), where the surface of the airfoil is parameterized with an NACA four-digit representation. Burgreen [56] applied this

approach in two and three dimensions; the boundary was represented with Bezier-Bernstien parameterization. Recently, Jameson and Reuther [20] applied this approach to airfoil optimization.

Another approach is to construct a set of rules by which the grid is moved after the initial grid is generated and then to differentiate these rules to obtain the grid sensitivity. This approach is used, for example, when the initial mesh is generated using a computer aided design [CAD] package. Taylor et al. [39] proposed a procedure for calculating grid sensitivity terms and for use in efficient grid regeneration. As the shape of the flow domain continuously changes as required by any shape optimization process, the mesh points in the domain must be properly adjusted in the design iterations to avoid the numerical errors induced by excessive mesh distortion. The requirement of mesh regridding distinguishes shape design optimization from other design-optimization applications. This procedure is used in the present 2-D study to obtain grid sensitivity. This method, which will be presented subsequently, is based on an "elastic membrane" analogy to represent the computational domain, with grid SD's calculated from a standard structural-analysis code by using the finite-element method.

A simple method for automatic mesh regridding can be established by introducing a set of basic displacement vectors $\bar{V}_k$ to describe the patterns by which the mesh is to be regridded. The relationship between the original mesh $\bar{X}_o$ and the regridded mesh $\bar{X}$ can then be expressed in the form of a linear combination of those basic displacement vectors and their associated weighting coefficients β_k as

$$\bar{X} = \bar{X}_o + \sum_{i=1}^{ndv} \triangle\beta_k \bar{V}_k \qquad (3.23)$$

where the weighting coefficients are taken to be the design variables. The vector $\bar{X}_o$ represents the initial mesh, and ndv is the number of design variables which is produced with any conventional mesh-generation code; $\triangle\beta_k$ is the change in β_k which produces the new mesh $\bar{X}$ from the initial mesh $\bar{X}_o$. In this case, the basic displacement vector

$\bar{V}_k$ is simply equal to the required mesh sensitivity vector $\left\{ \frac{d\bar{X}}{d\beta_k} \right\}$. That is, the grid SD's are calculated by differentiation of Eq. (3.23), which yields

$$\left\{ \frac{d\bar{X}}{d\beta_k} \right\} = \{\bar{V}_k\} \tag{3.24}$$

Note that the grid-sensitivity vectors $\{\bar{V}_k\}$ do not change when the design variables are changed, provided that the domain is always regridded by using Eq. (3.23) as the shape of the domain changes. Therefore, these grid SD's must be calculated once and then stored prior to the start of an aerodynamic optimization strategy; they can be reused as often as needed for grid SA, as well as for automatic mesh regeneration.

The basic displacement vectors $\bar{V}_k$ can be in any form as long as they are each independent. In structural shape design optimization, the elastic displacements induced by the boundary perturbations are commonly selected to represent the basic displacement vectors. In this way, the movement of the mesh points is governed by linear elasticity, which not only preserves the continuity of the mesh but also avoids any mesh overlapping. The same practice must be applied to aerodynamic shape optimization problems, in which an imaginary elastic medium is introduced to represent the computational domain.

More specifically, the basic displacement vectors can be generated by either the fictitious load method [75] or the prescribed displacement method [76]. The former method produces basic displacement vectors by applying one unit load at each node along the boundary in the direction along which the node is allowed to move. This concept is illustrated in Fig. 3.1 for a representative airfoil grid. The latter method, however, produces the basic displacement vectors by imposing a nonzero displacement (in response to a unit change in each design variable) along the varied boundary. This concept is illustrated in Fig. 3.2 for a representative airfoil grid. The fictitious load method is usually applied to cases in which the location of each node on the varied boundary is considered as a design variable, whereas the prescribed displacement method is applied in cases in which the shape of the boundary to be designed is parameterized.

In the following example, a NACA four-digit airfoil is used to demonstrate the application of the prescribed displacement method for mesh regridding in an aerodynamic shape-optimization environment. The profile of the NACA four-digit airfoil can be precisely represented by polynomials in terms of the maximum thickness T, the maximum camber C, and the location of maximum camber L as

$$y(x) = \begin{cases} f(x) + C(2Lx - x^2)/L^2 & x \le L \\ f(x) + C(1 - 2L + 2Lx - x^2)/(1 - L)^2 & x > L \end{cases} \tag{3.25}$$

where

$$f(x) = \pm 0.5T(0.2969\sqrt{x} - 0.126x - 0.3516x^2$$
$$+ 0.2843\, x^3 - 0.1015x^4) \tag{3.26}$$

and the $\pm$ in the expression for f(x) indicates positive for the upper surface of the airfoil, and negative for the lower surface.

Because the derivatives of the airfoil shape with respect to T, C, and L are continuous, small changes in T, C, and L will induce small changes in airfoil shape. Therefore, with the employment of a Taylor's series expansion, such a change in the airfoil shape can be expanded approximately into a linear function of ΔT, ΔC, and ΔL given as

$$y(x) = y_o(x) + \frac{\partial y_o(x)}{\partial T}\Delta T + \frac{\partial y_o(x)}{\partial C}\Delta C + \frac{\partial y_o(x)}{\partial L}\Delta L \tag{3.27}$$

where

$$\Delta T = T - T_o$$
$$\Delta C = C - C_o$$
$$\Delta L = L - L_o \tag{3.28}$$

Above, T_o, C_o, and L_o are the initial values of these three shape parameters associated with the initial airfoil shape $y_o(x)$ and the initial grid $\bar{X}_o$.

The derivatives $\frac{\partial v_o(x)}{\partial T}$, $\frac{\partial v_o(x)}{\partial C}$, and $\frac{\partial v_o(x)}{\partial L}$ in Eq. (3.27) represent special patterns that control the allowable changes in the airfoil's shape. The new mesh $\bar{X}$ can be defined in a form given by Eq. (3.23) as

$$\bar{X} = \bar{X}_o + \Delta T \cdot \bar{V}_1 + \Delta C \cdot \bar{V}_2 + \Delta L \cdot \bar{V}_3 \qquad (3.29)$$

where ΔT, ΔC, and ΔL are taken to be the design variables (or, equivalently, T, C, and L are the design variables through Eq. (3.28)). The basic displacement vectors $\bar{V}_1$, $\bar{V}_2$, and $\bar{V}_3$ can be obtained by the prescribed displacement method as previously discussed. These vectors are obtained numerically through implementation of a finite-element model, with each cell in the computational mesh considered as a plane stress quadrilateral element. A finite-element matrix equation can then be formed to solve for each basic displacement vector (i.e., the movements of all grid points) throughout the elastic membrane model of the domain, in response to the nonzero boundary movement that is specified through Eq. (3.28) for a unit change (or some other conveniently scaled change) in each design variable. The finite-element matrix equation is linear with a symmetric and banded coefficient matrix. This equation is, therefore, solved directly by a single LU factorization; this LU factorization is then reused for multiple solutions (i.e., one solution $\bar{V}_k$ for each design variable).

Equation (3.27) clearly represents a particular parameterization of the airfoil surface that will only closely approximate the NACA four-digit parameterization (defined by Eqs. (3.25) and (3.26)) if ΔT, ΔC, and ΔL are small. However, if remaining exactly within or close to the allowable shapes defined by the NACA 4-digit parameterization is not necessary during the design, then Eq. (3.27) is a valid (but different) parameterization of the airfoil shape, even for large ΔT, ΔC, and ΔL. Thus, this classic NACA four-digit airfoil is presented only as an example.
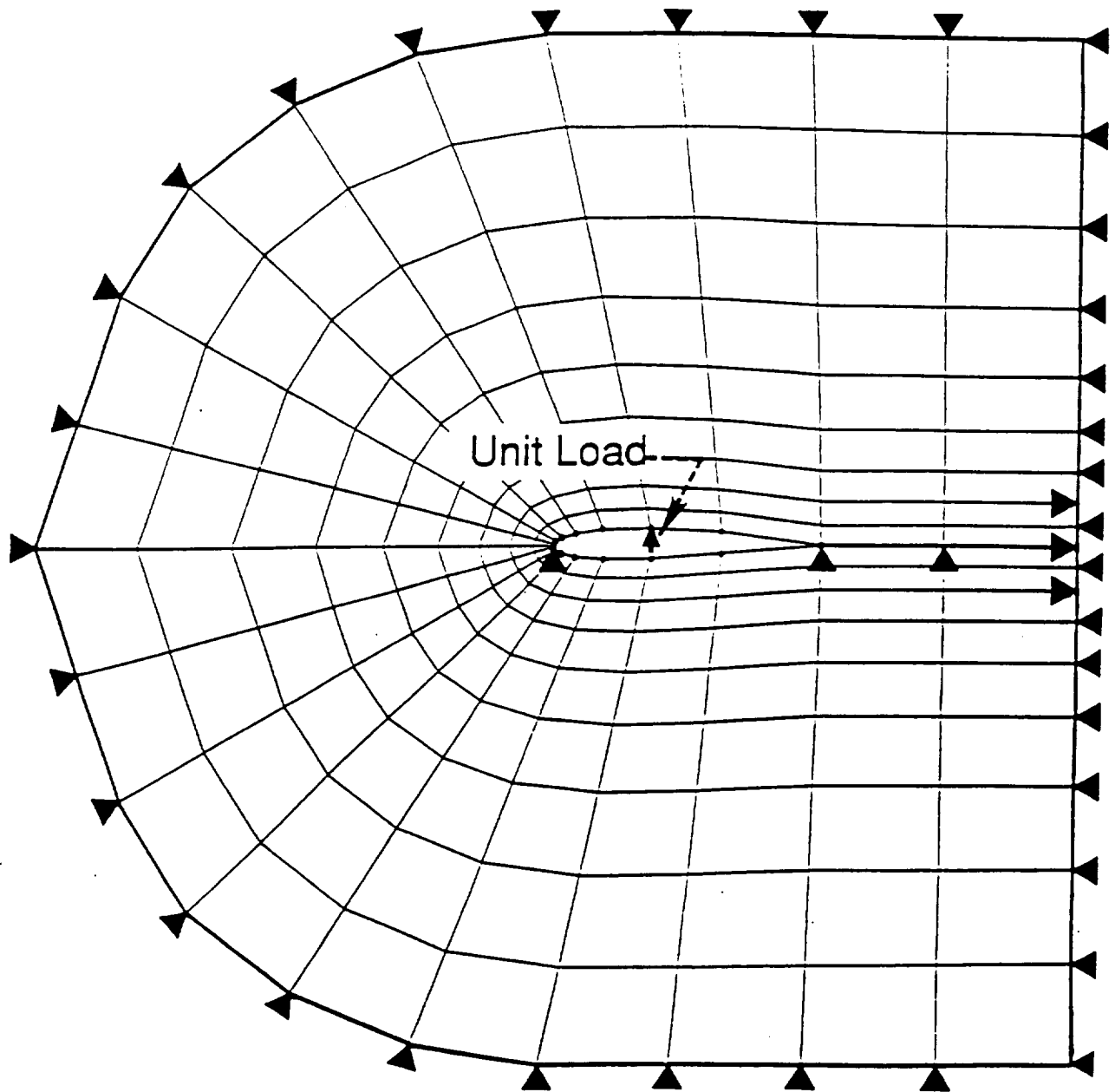
Unit Load

▲ Symbol Indicates Simple Support, Zero Displacement.
(Points Supported Included Leading And Trailing Edges,
Points In Wake, And All Points At The Far-Field Boundary)

Fig. 3.1 Illustration of elastic membrane representation of computational
domain with fictitious load method for computing grid sensitivity.

Prescribed Boundary Displacement

▲ Symbol Indicates Simple Support, Zero Displacement.
(Points Supported Included Leading And Trailing Edges,
Points In Wake, And All Points At The Far-Field Boundary)
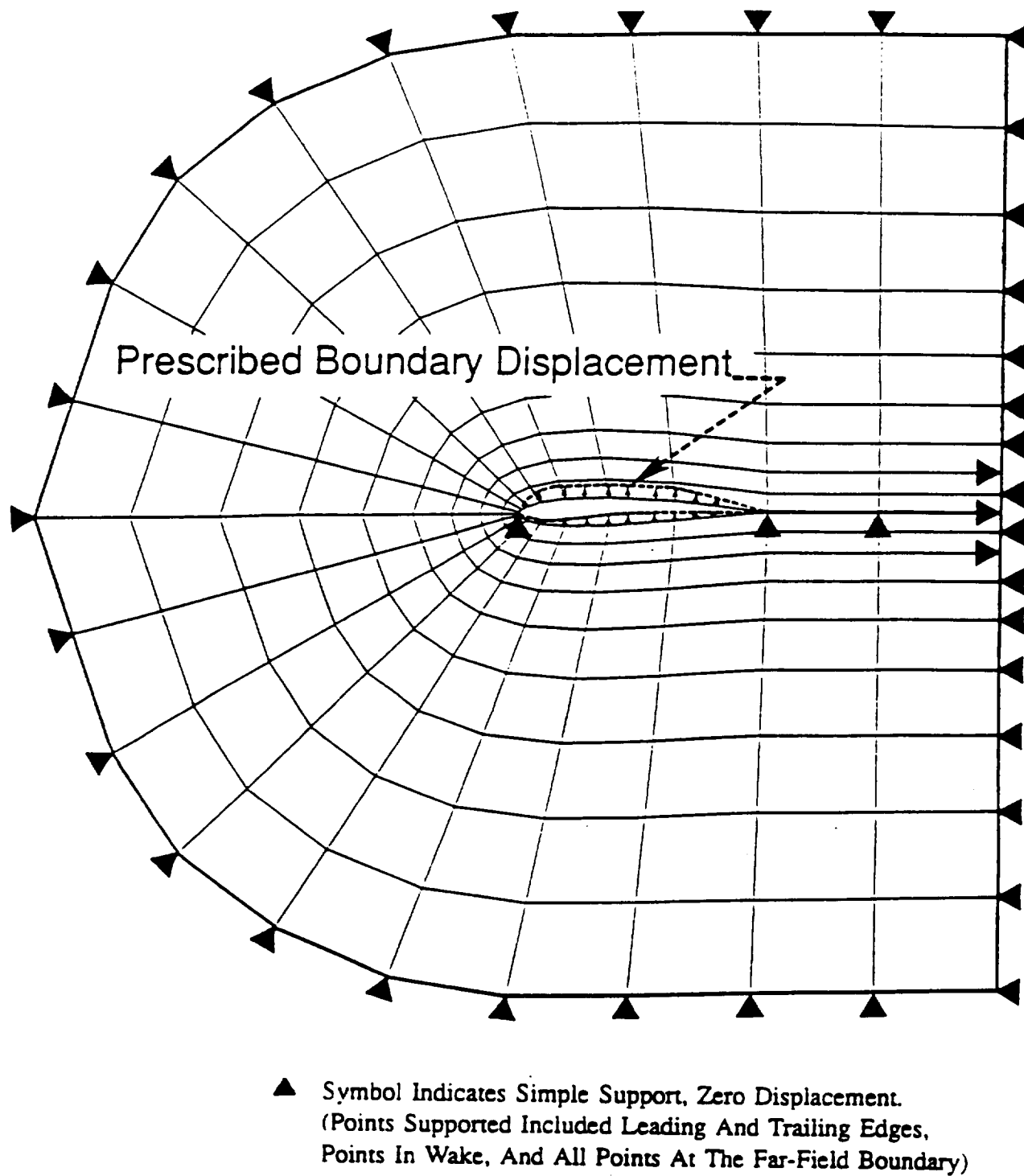
Fig. 3.2 Illustration of elastic membrane representation of computational domain with
prescribed boundary displacement method for computing grid sensitivity.

3.5 Algorithm for SD Calculation From a Marching Euler Code

In this section, a procedure is outlined to calculate SD's with the direct-differentiation approach in three dimensions. The algorithm is the same as that used to solve the nonlinear flow equations. This procedure is implemented in the computer code MARSEN, which was developed for this study, and was used in a gradient-based design-improvement study for the HSCT 24E. The procedure for calculating SD's with the adjoint-variable approach is given in Appendix C. Note that to solve for the adjoint vector, the marching must be done backwards (i.e., in the exact opposite direction to that of the flow).

The procedure for calculating SD's in three dimensions is a direct extension of the method in two dimensions. The residual equation in the ith cross plane is differentiated with respect to the kth component of the design variable vector $\vec{\beta}$ by using the implicit function theorem. Although the governing fluid equations are nonlinear in the state variables Q^*, the resulting sensitivity equations are linear in the sensitivity of the state variables $\left\{ \frac{dQ^*}{d\beta_k} \right\}$. The residual in the ith cross plane is written as a function of the state variables in the i, i-1, and i-2 cross planes, the grid coordinates $\bar{X}$, with explicit dependence on the design variable β_k:

$$\left\{ R_i \left(Q_i^*, Q_{i-1}^*, Q_{i-2}^*, \bar{X}, \beta_k \right) \right\} = \{0\} \tag{3.30}$$

Here, the subscripts j and k on the state variables Q^* are suppressed for simplicity. Differentiating Eq. (3.30) with respect to the design variable β_k, then the following equation results:

$$\left\{ \frac{dR_i}{d\beta_k} \right\} = \left[\frac{\partial R_i}{\partial Q_i} \right] \left\{ \frac{dQ_i^*}{d\beta_k} \right\} + \left[\frac{\partial R_i}{\partial Q_{i-1}} \right] \left\{ \frac{dQ_{i-1}^*}{d\beta_k} \right\} + \left[\frac{\partial R_i}{\partial Q_{i-2}} \right] \left\{ \frac{dQ_{i-2}^*}{d\beta_k} \right\}$$
$$+ \left[\frac{\partial R_i}{\partial \bar{X}} \right] \left\{ \frac{d\bar{X}}{d\beta_k} \right\} + \left\{ \frac{\partial R_i}{\partial \beta_k} \right\} = \{0\} \tag{3.31}$$

In Eq. (3.31), the vectors $\left\{ \frac{dQ_i^*}{d\beta_k} \right\}$, $\left\{ \frac{dQ_{i-1}^*}{d\beta_k} \right\}$, and $\left\{ \frac{dQ_{i-2}^*}{d\beta_k} \right\}$ are the sensitivities of the fluid variables with respect to the design variable β_k in the i, $i - 1$, and $i - 2$ cross planes. The

important point here is that upwind interpolation of the cell-centered values Q_i^* to the cell faces for evaluation of inviscid fluxes involves state variables in only the $i - 1$ and $i - 2$ cross planes because of the nature of inviscid, supersonic flow. The matrices $\left[\frac{\partial R_i}{\partial Q_i}\right]$, $\left[\frac{\partial R_i}{\partial Q_{i-1}}\right]$ and $\left[\frac{\partial R_i}{\partial Q_{i-2}}\right]$ that are the same Jacobian matrices that are discussed in the implicit formulation. The Jacobian matrix $\left[\frac{\partial R_i}{\partial X}\right]$ is sparse and banded. This Jacobian matrix is computed as $\left[\frac{\partial R}{\partial M}\right]\left[\frac{\partial M}{\partial X}\right]$, where M represents the metric terms and $\bar{X}$ represents the grid coordinates. Differentiation of the residual expression with respect to metric terms is straightforward and is not discussed here. The vector $\left\{\frac{\partial R_i}{\partial \beta_k}\right\}$ accounts for explicit dependencies, if any, of the residual vector R_i on the design variable β_k. Equation (3.31) can be written in standard form as

$$-\left[\frac{\partial R_i}{\partial Q_i}\right]\left\{\frac{dQ_i^*}{d\beta_k}\right\} = \left[\frac{\partial R_i}{\partial Q_{i-1}}\right]\left\{\frac{dQ_{i-1}^*}{d\beta_k}\right\} + \left[\frac{\partial R_i}{\partial Q_{i-2}}\right]\left\{\frac{dQ_{i-2}^*}{d\beta_k}\right\}$$
$$\left[\frac{\partial R_i}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_i}{\partial \beta_k}\right\} \tag{3.32}$$

The sensitivities of the state variables in the $i - 1$ and $i - 2$ cross planes $\left(\left\{\frac{dQ_{i-1}^*}{d\beta_k}\right\}, \left\{\frac{dQ_{i-2}^*}{d\beta_k}\right\}\right)$ are known when sensitivities of the state variables in the ith cross plane are solved with a space-marching algorithm in fully supersonic flow. Equation (3.32) is linear in the unknown $\left\{\frac{dQ_i^*}{d\beta_k}\right\}$. By casting this equation in incremental iterative form the following equation results:

$$-\left[\widetilde{\frac{\partial R_i}{\partial Q_i}}\right]\left\{{}^m\Delta\frac{dQ_i}{d\beta_k}\right\} = \left[\frac{\partial R_i}{\partial Q_i}\right]\left\{\frac{dQ_i}{d\beta_k}\right\}^m + \left[\frac{\partial R_i}{\partial Q_{i-1}}\right]\left\{\frac{dQ_{i-1}^*}{d\beta_k}\right\}$$
$$+ \left[\frac{\partial R_i}{\partial Q_{i-2}}\right]\left\{\frac{dQ_{i-2}^*}{d\beta_k}\right\} + \left[\frac{\partial R_i}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_i}{\partial \beta_k}\right\} \tag{3.33a}$$

$$\left\{\frac{dQ_i}{d\beta_k}\right\}^{m+1} = \left\{{}^m\Delta\frac{dQ_i}{d\beta_k}\right\} + \left\{\frac{dQ_i}{d\beta_k}\right\}^m$$

$$m = 1, 2, 3, \dots \tag{3.33b}$$

In Eq. (3.33), the left-hand-side matrix operator $\left[\widetilde{\frac{\partial R_i}{\partial Q_i}}\right]$ is the approximation of convenience of the matrix $\left[\frac{\partial R_i}{\partial Q_i}\right]$ and is chosen such that it makes the iterative process convergent. For the present study, the first-order upwind discretization of the Jacobian matrix is used as the matrix operator. A time term which acts as an under-relaxation parameter is added to the left-hand-side matrix operator. Equation. (3.33) is solved for each cross plane, and the vector $\left\{\frac{dQ_i^*}{d\beta_k}\right\}$ is calculated over the whole domain. After this complete vector is known, the sensitivity of the system response of interest with respect to the design variable can be computed with Eq. (3.2).

Chapter 4

COMPUTATIONAL RESULTS

In this chapter, the SD results in two dimensions are given in separate sections for two sample airfoil problems: subsonic low-Reynolds-number laminar flow and transonic high-Reynolds-number turbulent flow. Sample 3-D SD results are given for geometric and non-geometric design variables in separate subsections.

4.1 Subsonic Airfoil, Low Reynolds Number Laminar Flow

The first problem is subsonic low-Reynolds-number, constant-viscosity laminar flow over an NACA 1406 airfoil. Flow is considered at a freestream Mach number M_∞ = 0.6, an angle of attack α = 1.0°, and a Reynolds number Re = 5.0 × 10^3. A C-mesh computational grid of 257 × 65 points is used, with the "lift-corrected" far-field boundary placed five chords from the airfoil; points are clustered near the airfoil surface to assist with the resolution of gradients in this vicinity. The cell-centered finite-volume formulation method with higher upwind differencing for the inviscid terms and central differencing for viscous terms is used. The spatially split approximate factorization algorithm is used to achieve the converged (i.e., the average global error is reduced to machine-zero) steady-state solution $\{Q^*\}$ of the discrete, nonlinear flow equations. Figure 4.1 is a plot of the computed steady-state pressure coefficient C_p on the surface of the airfoil. The computed lift, drag, and pitching moment coefficients obtained are C_L = 0.18148, C_D = 0.41703 E-01, and C_M = − 0.23718 E-01.

The SD's of C_L, C_D, and C_M are computed with respect to six independent design

variables: airfoil maximum thickness T; airfoil maximum camber C; location of maximum camber L; angle of attack α; freestream Mach number M_∞; and Reynolds number Re. The three design variables related to geometric shape (T, C, and L) are parameters that together with well-known analytical expressions (given, for example, in Ref. [39]) define the x and y coordinates on the surface (and, hence, the shape) of the NACA four–digit airfoil. The SD's are computed with three methods: the direct-differentiation method; the adjoint-variable method; and the "brute-force" finite-difference method. Application of these three methods is described subsequently in greater detail; comparisons of the computational results are summarized in Table 4.1. For the direct-differentiation and the adjoint-variable method, noted that the direct-solver approach was abandoned because of storage restrictions. In this case, ("in core") storage required by the banded matrix far exceeded the 40–megaword storage limit placed on the standard Cray-2 computer queue.

For the direct-differentiation method, SD's are calculated through the iterative solution of the incremental form (i.e., Eqs. (3.15), (3.16), and (3.17)) of six large systems of linear equations (one system for each of the six design variables considered here). The well-known spatially split approximate factorization algorithm [66] is used, with a constant Courant number of 45 (i.e., local time stepping is used). This Courant number was determined by numerical experimentation to be approximately the optimum for computational efficiency for this sample problem. An eight- order-of-magnitude reduction in the average global error is the specified convergence criterion for solving each of the six linear systems; an average of 683 iterations is required in each case to achieve this convergence criterion.

For the adjoint-variable method, SD's are calculated through the iterative solution of the incremental form (i.e., Eqs. (3.18) and (3.19)) of three large systems of linear equations, one system for each of the three system responses considered here. Again the approximate factorization algorithm is used, and a constant Courant number of 45 is determined to be the optimum. In this case, an average of 1743 iterations is required to

obtain an eight-order-of-magnitude average global error reduction, which is the required convergence criterion for each of these three linear system solutions.

In application of the "brute-force" finite-difference method, central finite differencing is used, with a forward and backward perturbation of each design variable ($\Delta \beta_k = \pm 5.0E - 06 \times \beta_k$). Machine-zero converged, steady-state solutions of the discrete nonlinear flow equations are obtained for each forward and backward perturbation of each design variable. Thus, for six design variables a total of 12 solutions to the nonlinear flow equations are produced. The approximate factorization algorithm is again used to solve the flow equations; to reduce computational work during these computations, the LU-factored block-tridiagonal systems are stored and are reused for 10 iterations; after 10 iterations these terms are reevaluated. (See Ref. [65] for additional details in regard to this strategy, which was shown with numerical studies to be near optimum.)

The SD's calculated with the direct-differentiation method agree closely with those computed with the adjoint-variable method. However, the computational work required by the latter method (in which a total of three linear systems are solved) exceeds that of the former method (in which a total of six linear systems are solved). In addition, the convergence rates obtained with the latter method were significantly slower than those obtained with the former method in this sample problem. The SD's obtained by using finite differencing also agree closely with those obtained from the other two methods. In all comparisons, the finite-difference method was much more costly computationally than either the direct-differentiation or the adjoint-variable method.

Fig. 4.1 Chordwise distribution of surface pressure coefficient NACA
1406 airfoil, $M_\infty = 0.6$; $\alpha = 1.0°$; $Re = 5 \times 10^3$; laminar flow.

Table 4.1 Summary of Computational Results for NACA 1406 Airfoil:
Subsonic Low-Reynolds-Number Laminar Flow Sample Problem

Solution method	Total CPU time (Secs)[*]	Design variable β_k	$\dfrac{dC_L}{d\bar\beta}$	$\dfrac{dC_D}{d\bar\beta}$	$\dfrac{dC_M}{d\bar\beta}$
Direct-Differentiation method, approximately factored incremental scheme	458	T	-1.392 E+00	+2.019 E-01	+1.805 E-01
		C	+6.583 E+00	+7.583 E-02	-2.240 E+00
		L	-1.154 E-02	+5.544 E-05	-2.122 E-02
		α	+6.122 E+00	+9.181 E-02	-3.168 E-02
		M_∞	+5.428 E-03	+1.628 E-02	-4.732 E-03
		Re	+5.958 E-06	-4.912 E-06	-6.564 E-07
Adjoint -variable method, approximately factored incremental scheme	579	T	-1.392 E+00	+2.019 E-01	+1.805 E-01
		C	+6.583 E+00	+7.583 E-02	-2.240 E+00
		L	-1.154 E-02	+5.544 E-05	-2.122 E-02
		α	+6.122 E+00	+9.181 E-02	-3.168 E-02
		M_∞	+5.428 E-03	+1.628 E-02	-4.732 E-03
		Re	+5.958 E-06	-4.912 E-06	-6.564 E-07
"Brute -force" finite difference method	7404	T	-1.392 E+00	+2.019 E-01	+1.805 E-01
		C	+6.583 E+00	+7.583 E-02	-2.240 E+00
		L	-1.154 E-02	+5.548 E-05	-2.122 E-02
		α	+6.122 E+00	+9.181 E-02	-3.168 E-02
		M_∞	+5.426 E-03	+1.628 E-02	-4.732 E-03
		Re	+5.958 E-06	-4.912 E-06	-6.564 E-07

[*]All calculations performed on Cray-2 computer.

4.2 Transonic Airfoil, High Reynolds Number Turbulent Flow

The second sample problem is transonic high-Reynolds number turbulent flow over an NACA 1406 airfoil. The variation of the molecular viscosity with temperature is computed with Sutherland's law, and turbulence is simulated with the well-known algebraic model of Baldwin and Lomax [78]. The flow is considered at a freestream Mach number $M_\infty = 0.8$, an angle of attack $\alpha = 1.0°$, and a Reynolds number $Re = 5.0 \times 10^6$. A C-mesh with 257×65 grid points is again used with the lift-corrected far-field boundary placed five chords from the airfoil; clustering of points near the surface is tighter in the present example than in the previous example because of the higher Reynolds number. The cell-centered finite-volume formulation method with higher upwind differencing for the inviscid terms and central differencing for viscous terms is used. The spatially split approximate factorization algorithm is used to achieve a machine-zero converged, steady-state solution. Figure 4.2 is a plot of the computed steady-state pressure coefficient C_p on the surface of the airfoil, and Fig. 4.3 is a complete contour plot of the static pressure, which clearly shows the presence of a shock wave on the suction surface of the airfoil. The computed lift, drag, and pitching moment coefficients are $C_L = 0.41662$, $C_D = 0.77501$ E-02, and $C_M = -0.45633$ E-01.

The SD's of C_L, C_D, and C_M are computed with respect to the same six independent design variables previously considered. The direct-differentiation, the adjoint-variable, and the "brute-force" finite-difference methods are also applied in computing these SD's. However, for the direct-differentiation and adjoint-variable methods, laminar and turbulent viscosities are assumed to be constant with respect to the field variables $\{Q^*\}$ and the computational grid $\{\bar{X}\}$. That is, in the analytical construction of all derivatives (including the Jacobian matrices $\left[\frac{\partial R}{\partial Q}\right]$ and $\left[\frac{\partial R}{\partial X}\right]$), which are used to calculate the SD's, both laminar and turbulent viscosities are constant. For this reason, the direct-differentiation and the adjoint-variable methods cannot give SD's that are exact, consistently discrete forms. Thus, the results from the "brute-force" finite-difference

procedure are considered to be more accurate in this example. This approximation is made because of the complexity involved in the consistent treatment the derivatives of the turbulent viscosity. In fact, a fully consistent treatment of these terms is not possible at points where this turbulence model is not continuously differentiable. Application of the three methods is described subsequently in greater detail. Comparison of the computational results are summarized in Table 4.2.

For the direct-differentiation and adjoint-variable methods, the SD's are computed with the spatially split approximate factorization algorithm to iteratively solve in incremental form the required linear systems that have been described. With both methods, a constant Courant number of 30 is numerically determined as the optimum for the computations. In all cases an eight–order-of-magnitude reduction in the average global error is enforced for convergence. For the direct-differentiation method, an average of 1619 iterations is needed to achieve convergence; for the adjoint-variable method, an average of 1798 iterations is required. Finally, the "brute-force" finite-difference method is applied here in a manner identical to that described in the previous sample problem.

The SD's calculated with the direct–differentiation method and with the adjoint-variable method agree well, as expected. In addition, the total computational cost of the direct-differentiation method is approximately twice the cost of the adjoint-variable method. This result is expected because with the direct-differentiation method six linear systems are solved compared with only three for the adjoint-variable method. The SD's calculated using the method of finite differences are compared with those from the other two methods; some discrepancy occurs in the results because of the aforementioned neglected consistent treatment of the viscosity terms. For the most part, the agreement between these calculated derivatives is good. The most significant discrepancy is noted in the SD's of C_L with respect to maximum airfoil thickness T, where the derivatives differ by a factor of approximately three to four. However, this SD is smaller in magnitude than

the largest derivatives. As in the first sample problem, the "brute-force" finite-difference method is much more costly computationally than either the direct-differentiation or the adjoint-variable method.

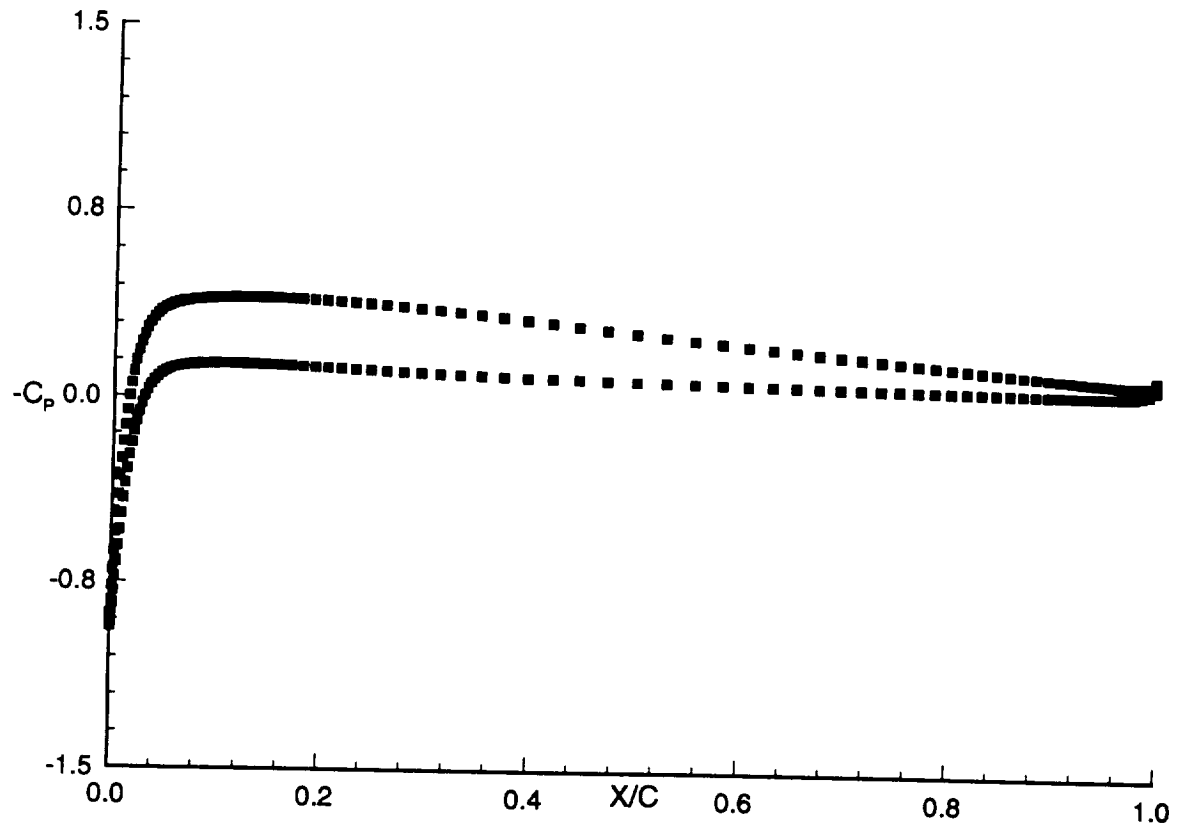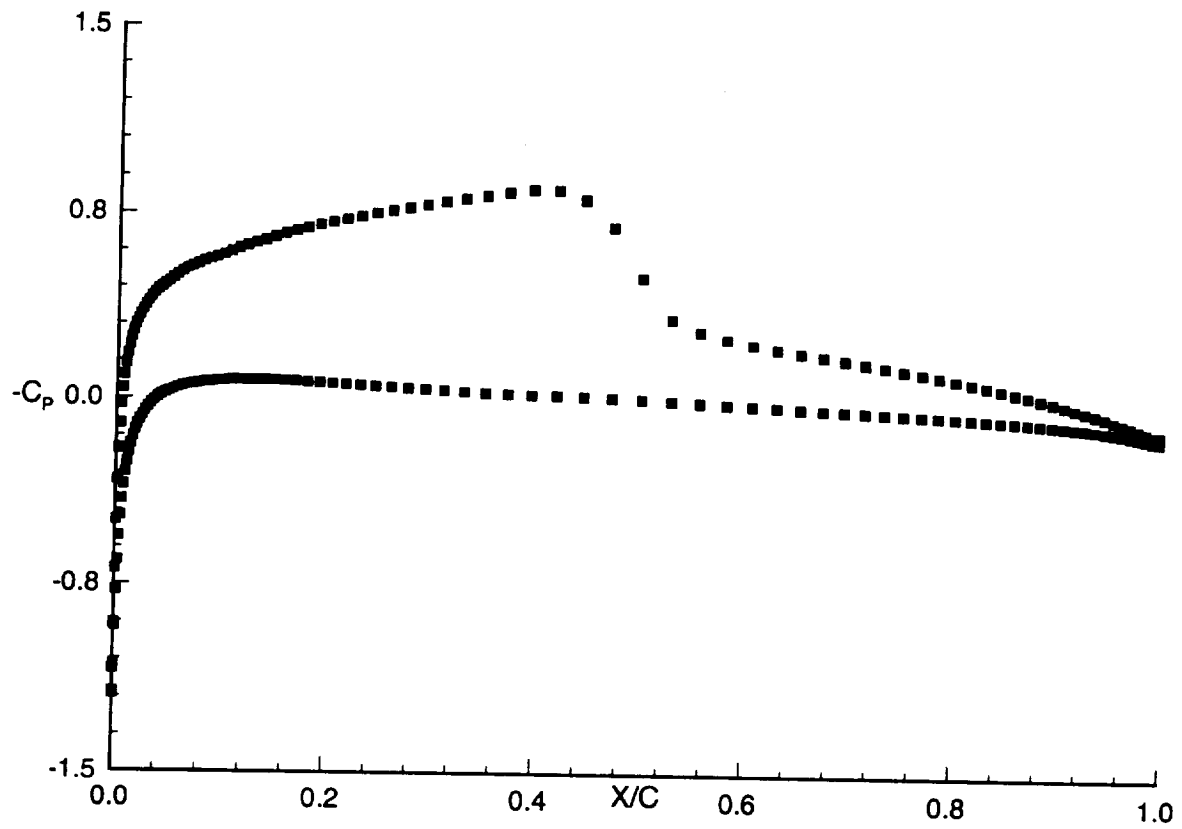Fig. 4.2 Chordwise distribution of surface pressure coefficient. NACA 1406 Airfoil; $M_\infty = 0.8$; $\alpha = 1.0°$; Re = 5×10^6; turbulent flow.
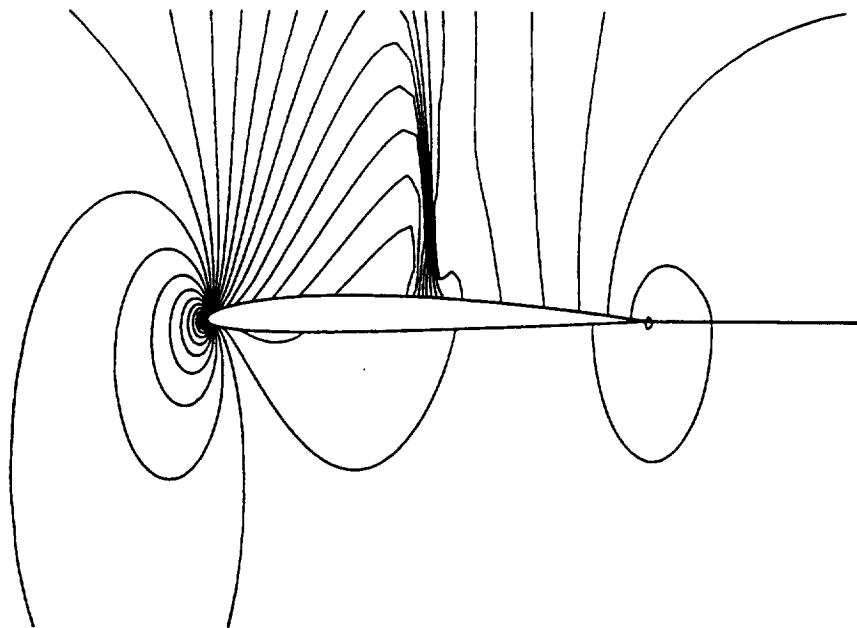
Fig. 4.3 Static pressure contour plot. NACA 1406 airfoil,
$M_\infty = 0.8$; $\alpha = 1.0°$; $Re = 5 \times 10^6$; turbulent flow.

Table 4.2 Summary of Computational Results for NACA 1406 Airfoil:
Transonic High-Reynolds-Number Turbulent Flow Sample Problem

Solution method	Total CPU time (Secs)[*]	Design variable β_k	$\dfrac{dC_L}{d\bar{\beta}}$	$\dfrac{dC_D}{d\bar{\beta}}$	$\dfrac{dC_M}{d\bar{\beta}}$
Direct-Differentiation, approximately factored incremental scheme	1052	T	+2.275 E-01	+2.654 E-01	-3.124 E-01
		C	+1.942 E+01	+6.511 E-01	-5.516 E+00
		L	+1.338 E-01	-1.151 E-02	-5.589 E-02
		α	+1.198 E+01	+4.200 E-01	-4.675 E-01
		M_∞	+1.772 E+00	+1.921 E-01	-5.430 E-01
		Re	+4.145 E-09	-4.881 E-10	-4.397 E-10
Adjoint -variable method, approximately factored incremental scheme	586	T	+2.275 E-01	+2.654 E-01	-3.124 E-01
		C	+1.942 E+01	+6.511 E-01	-5.516 E+00
		L	+1.338 E-01	-1.151 E-02	-5.589 E-02
		α	+1.198 E+01	+4.200 E-01	-4.675 E-01
		M_∞	+1.772 E+00	+1.921 E-01	-5.430 E-01
		Re	+4.145 E-09	-4.881 E-10	-4.397 E-10
Brute-force finite-difference method	8526	T	+7.919 E-01	+2.744E-01	-4.153 E-01
		C	+2.063 E+01	+6.776 E-01	-5.770 E+00
		L	+1.107 E-01	-1.174 E-02	-5.350 E-02
		α	+1.299 E+01	+4.346 E-01	-6.328 E-01
		M_∞	+2.040 E+00	+1.969 E-01	-5.972 E-01
		Re	-1.185 E-09	-2.829 E-10	+1.497 E-10

[*]All calculations performed on a Cray-2 computer.

4.3 Comparison of SD Results in Three Dimensions

The 3-D Euler equations are solved here for a fully supersonic flow with the space-marching method described in Chap. 2. The method is an upwind cell-centered finite-volume scheme that is higher-order accurate (second-order streamwise and third-order in the cross plane) and fully conservative in all directions, including the streamwise (marching) direction. The method is locally time iterative in each cross plane with a spatially split approximate-factorization approach. The Mach 2.4 filleted wing-body surface definition was processed with the method given in Ref. [79] and a volume grid subsequently generated as in Ref. [80]. Figure 4.4 is a view of the HSCT 24E (High-Speed Civil Transport) filleted wing-body configuration, including the wake portion of the computational grid.

4.3.1 Geometric Design Variables

Comparisons are made of the SD's obtained with central finite differencing (SD_{FD}) and the IIM for several geometric variables. The geometric design variables are those variables that define the surface of the HSCT 24E wing. Details of the wing-geometry parameterization are given in Appendix D. Grid generation and grid sensitivity for the present study are obtained by automatically differentiating the surface/volume-grid generator (Refs. [79, 80]). The flight conditions chosen are M_∞= 2.4, $\alpha = 1°$, $\beta = 0°$.

The geometric SD results are computed on a half-space grid (37 streamwise × 49 circumferential × 15 normal points) with a symmetry plane at y = 0; some forces, moments, and SD's are not balanced by their images and, therefore, do not vanish. These nonvanishing components do not affect the geometric SD comparisons for the six-component force and moment coefficient $\left(C_x, C_y, C_z, C_{M_x}, C_{M_y}, C_{M_z}\right)$ SD's with respect to the geometric design variables. In obtaining the SD_{FD}, analysis solutions at design-variable perturbations of approximately 10^{-5} from the baseline were run from restart solution files and converged to a relative residual reduction of 10^{-11}. This process results in an appreciable time savings for obtaining the SD_{FD}, at least from the present CFD

algorithm and code. The spatially split approximate-factorization algorithm is used to solve the sensitivity equation in each cross plane with IIM. A constant Courant number of 10 is used for the computations. In obtaining the SD's via the IIM (SD_{QA}), the relative derivative-residual reduction was done to several levels: 10^{-3} (3 orders of magnitude (OM)), 10^{-7} (7 OM), and 10^{-11} (11 OM). Comparisons are shown for both accuracy and computational efficiency.

Six SD's are compared with respect to three wing-section thickness ratios (t/C) in Table 4.3. This table has five parts: part (a) gives the 18 SD_{QA}; parts (b), (c), and (d) show the 18 ratios (SD_{FD}/SD_{QA}) for 3, 7, and 11 OM, respectively; and part (e) gives computational time comparisons. Table 4.3(a) shows that these derivatives range in size over nearly 3 OM and are both positive and negative. Tables 4.3(b)-(d) show that the SD_{QA} agree with the SD_{FD} to between three and four significant figures. Table 4.3(e) shows the computation of SD_{QA} to be 1.5 to 2 times faster than the computation of the efficient SD_{FD} (i.e., with restarts, central finite-difference time is about 2.3 rather than 6 times a baseline analysis solution time). The speed-up depends on the SD accuracy required and the analysis code convergence performance from restarts.

Tables 4.4, 4.5, and 4.6 compare similar SD results for sample section twist, camber, and flap-deflection geometric variables, respectively. For these cases, however, only the 11 OM SD_{QA} comparisons are shown. Again, these derivatives vary over several OM in size; however, agreement with the SD_{FD} remains better than to three significant figures; the derivatives are obtained about 1.5 times faster than those derivatives obtained with the best SD_{FD} computation.

Comparison of the six SD's with respect to three wing planform variables is shown in Table 4.7. Here, SD comparisons are shown at all three SD_{QA} convergence levels. The SD_{QA} agree with the SD_{FD} to about four significant figures; in addition, they are obtained faster with the IIM.

4.3.2 Nongeometric Design Variables

As a consequence of using the IIM, the linear sensitivity equations are solved for the SD's of the field variables in each cross plane with the identical space-marching algorithm that is used to solve the nonlinear flow equations. The computational grid used for this study ($37 \times 121 \times 15$, with 37 points in streamwise direction, with 121 circumferential direction, and 15 points in the normal direction) is different from the grid used to study the geometric design variables. Force and moment coefficients for the flight conditions $M_\infty = 2.4$, $\alpha = 0°$, $\beta = 0°$ are shown in Table 4.8. The SD's of six output functions $(C_x, C_y, C_z, C_{M_x}, C_{M_y}, C_{M_z})$ with respect to Mach, Alpha, and Beta are given in Table 4.9(a). Calculated SD ratios, (forward finite differences with a perturbation size, $\Delta\beta_k = 1.E-05$ to quasi-analytical derivatives) are shown in Table 4.9(b); these ratios are seen to be unity to four significant figures. Table 4.9(c) shows computational time comparisons for the calculation of SD's with using both forward finite differences and the quasi-analytical IIM; all times are given in terms of a baseline time. The measure of convergence levels used for the solutions of the nongeometric design variables is given in the footnote to Table 4.9(c). Three nonlinear flow solutions, which correspond to the perturbed flow conditions, are obtained by using the freestream conditions as the initial guess. The computational cost of the finite-difference method is approximately seven times greater compared with that for quasi-analytical method.
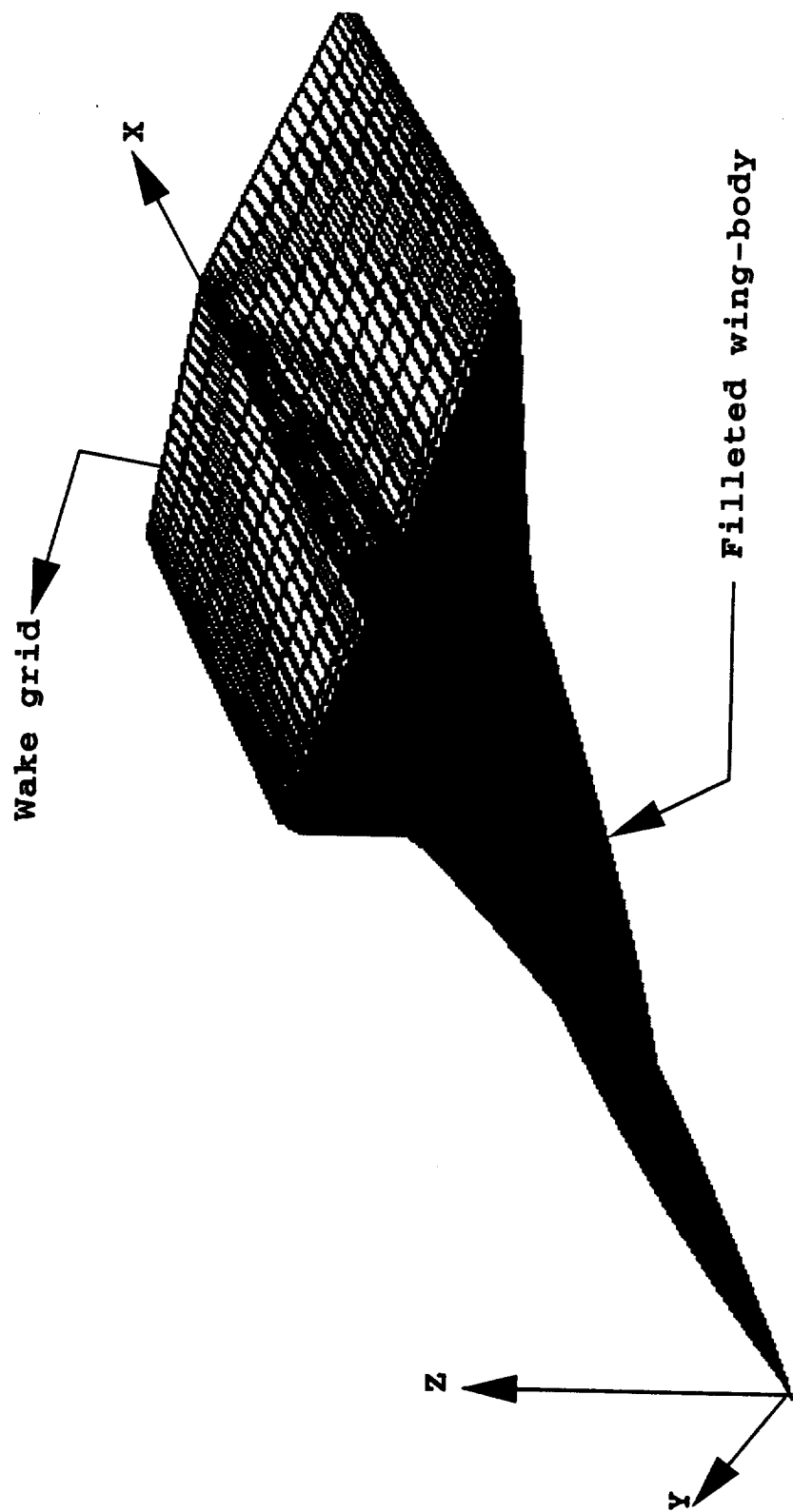
Fig. 4.4 HSCT 24E filletted wing-body configuration.

Table 4.3 (a) Geometric Section Thickness SD's of Force and Moment Coefficients With Quasi-Analytical Incremental Iterative Method (QAIIM) for HSCT 24E at M_∞ = 2.4, α = 1°, and β = 0°

SD_{QA}	Scaled design variables $\bar{\beta}$		
	Root t/c	Break t/c	Tip t/c
$\frac{dC_x}{d\beta_k}$	+3.8635 E-04	+2.8663 E-04	+3.3805 E-05
$\frac{dC_y}{d\beta_k}$	−2.4830 E-05	−2.8052 E-04	−2.0875 E-05
$\frac{dC_z}{d\beta_k}$	+4.5475 E-04	+6.1267 E-05	+4.7231 E-06
$\frac{dC_{M_x}}{d\beta_k}$	+2.0925 E-05	−1.2866 E-05	+6.1225 E-07
$\frac{dC_{M_y}}{d\beta_k}$	+3.4438 E-06	−5.7055 E-06	−2.0632 E-06
$\frac{dC_{M_z}}{d\beta_k}$	+1.7229 E-04	−7.6030 E-05	−1.3698 E-05

Table 4.3 (b) Geometric Section Thickness SD Ratios ($\frac{\text{Finite Difference}}{QA}$)

$\frac{SD_{FD}}{SD_{QA}}$	Design variables $\bar{\beta}$		
	Root t/c	Break t/c	Tip t/c
$\frac{dC_x}{d\beta_k}$	1.0000	0.9999	1.0000
$\frac{dC_y}{d\beta_k}$	1.0054	0.9997	1.0004
$\frac{dC_z}{d\beta_k}$	0.9995	0.9999	1.0011
$\frac{dC_{M_x}}{d\beta_k}$	0.9984	1.0005	1.0023
$\frac{dC_{M_y}}{d\beta_k}$	1.0431	1.0018	1.0007
$\frac{dC_{M_z}}{d\beta_k}$	0.9997	0.9996	1.0003

(Reduction of 3 OM)

Table 4.3 (c) Geometric Section Thickness SD Ratios ($\frac{\text{Finite Difference}}{\text{QA}}$)

$\frac{SD_{FD}}{SD_{QA}}$	Design variables $\bar{\beta}$		
	Root t/c	Break t/c	Tip t/c
$\frac{dC_x}{d\beta_k}$	0.9999	0.9999	0.9999
$\frac{dC_y}{d\beta_k}$	1.0000	0.9999	1.0000
$\frac{dC_z}{d\beta_k}$	0.9999	0.9999	1.0000
$\frac{dC_{M_x}}{d\beta_k}$	1.0000	0.9999	1.0000
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0000	1.0000
$\frac{dC_{M_z}}{d\beta_k}$	1.0000	0.9999	0.9999

(Reduction of 7 OM)

Table 4.3 (d) Geometric Section Thickness SD Ratios ($\frac{\text{Finite Difference}}{\text{QA}}$)

$\frac{SD_{FD}}{SD_{QA}}$	Design variables $\bar{\beta}$		
	Root t/c	Break t/c	Tip t/c
$\frac{dC_x}{d\beta_k}$	0.9999	0.9999	0.9999
$\frac{dC_y}{d\beta_k}$	1.0000	0.9999	1.0000
$\frac{dC_z}{d\beta_k}$	0.9999	1.0000	0.9999
$\frac{dC_{M_x}}{d\beta_k}$	1.0000	0.9999	1.0000
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0000	1.0000
$\frac{dC_{M_z}}{d\beta_k}$	1.0000	0.9999	0.9999

(Reduction of 11 OM)

Table 4.3 (e) Geometric Section-Thickness SD Computational-Time Comparisons

Solution Method	Number of solutions	Ratio
Baseline	1	1.000*
Central finite differencing	6	1.289
Quasi-analytical (3 OM)	3	0.2032
Quasi-analytical (7 OM)	3	0.2817
Quasi-analytical (11 OM)	3	0.3714

* Baseline solution run time for $\left(\frac{R^n_{rms}}{R^1_{rms}}\right)$ reduction to $\epsilon \equiv 10^{-11}$ on Cray-2 is 152 sec.

Table 4.4 (a) Geometric Twist SD of Force and Moment Coefficients
With QAIIM for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

SD_{QA}	Scaled design variables $\bar{\beta}$		
	Root twist	Break twist	Tip twist
$\frac{dC_x}{d\beta_k}$	-3.6909 E-04	+2.3174 E-05	-1.7165 E-07
$\frac{dC_y}{d\beta_k}$	+5.3123 E-03	-1.0226 E-04	-1.7900 E-06
$\frac{dC_z}{d\beta_k}$	+4.8539 E-03	-1.2541 E-03	-5.6965 E-06
$\frac{dC_{M_x}}{d\beta_k}$	+1.0684 E-04	-1.3584 E-04	-1.1203 E-06
$\frac{dC_{M_y}}{d\beta_k}$	-1.9188 E-03	+3.5747 E-04	+2.1336 E-06
$\frac{dC_{M_z}}{d\beta_k}$	+1.8410 E-03	-3.6119 E-05	-1.060 E-06

Table 4.4 (b) Geometric Twist SD Ratios ($\frac{\text{Finite Difference}}{\text{QA}}$) Except Terms of $O(\epsilon)$

$\frac{SD_{FD}}{SD_{QA}}$	Design variables $\bar{\beta}$		
	Root twist	Break twist	Tip twist
$\frac{dC_x}{d\beta_k}$	0.9999	1.0000	a
$\frac{dC_y}{d\beta_k}$	1.0000	0.9999	0.9999
$\frac{dC_z}{d\beta_k}$	1.0000	1.0000	1.0007
$\frac{dC_{M_x}}{d\beta_k}$	1.0000	1.0000	0.9999
$\frac{dC_{M_y}}{d\beta_k}$	1.0000	1.0000	1.0013
$\frac{dC_{M_z}}{d\beta_k}$	0.9999	0.9998	1.0000

[a] Ratio for extremely small quantities is meaningless.

Table 4.4 (c) Computational Time Comparisons

Solution Method	Number of solutions	Ratio
Baseline	1	1.000*
Central finite differencing	6	1.0755
Quasi-analytical	3	0.3141

* See note at Table 4.3.

Table 4.5 (a) Geometric Camber Surface SD of Force and Moment Coefficients With QAIIM for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

SD_{QA}	Scaled design variables $\bar{\beta}$		
	Root C	Break C	Tip C
$\frac{dC_x}{d\beta_k}$	-6.7160 E-06	+1.6566 E-05	-9.7360 E-08
$\frac{dC_y}{d\beta_k}$	2.4396 E-05	-3.0371 E-05	+7.3377 E-08
$\frac{dC_z}{d\beta_k}$	+6.1329 E-05	-7.8495 E-05	+1.3783 E-06
$\frac{dC_{M_x}}{d\beta_k}$	-7.9197 E-06	-9.2387 E-06	+3.4155 E-07
$\frac{dC_{M_y}}{d\beta_k}$	-6.7634 E-05	+6.4016 E-07	-4.6827 E-07
$\frac{dC_{M_z}}{d\beta_k}$	+1.0487 E-05	-8.5257 E-06	+3.8453 E-08

Table 4.5 (b) Geometric Camber Surface SD Ratios $\left(\frac{\text{Finite Difference}}{\text{QA}}\right)$ Except Terms of $O(\epsilon)$

$\frac{SD_{FD}}{SD_{QA}}$	Design variables $\bar{\beta}$		
	Root C	Break C	Tip C
$\frac{dC_x}{d\beta_k}$	0.9999	1.0000	a
$\frac{dC_y}{d\beta_k}$	0.9999	0.9999	a
$\frac{dC_z}{d\beta_k}$	0.9999	1.0000	1.0003
$\frac{dC_{M_x}}{d\beta_k}$	1.0000	1.0000	1.0003
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0000	1.0003
$\frac{dC_{M_z}}{d\beta_k}$	0.9999	0.9999	a

[a] See note at Table 4.4.

Table 4.5 (c) Geometric Camber Surface SD Computational-Time Comparisons

Solution method	Number of solutions	Ratio
Baseline	1	1.000*
Central finite differencing	6	0.883
Quasi-analytical	3	0.3084

* See note at Table 4.3.

Table 4.6 (a) Geometric Flap-Deflection SD of Force and Moment Coefficients With QAIIM for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

SD_{QA}	Scaled design variables $\bar{\beta}$			
	Flap I	Flap II	Flap III	Flap IV
$\frac{dC_x}{d\beta_k}$	+7.7336 E-06	+5.5417 E-06	+7.2944 E-08	+7.3339 E-07
$\frac{dC_y}{d\beta_k}$	-6.5184 E-06	+2.3167 E-05	-4.4830 E-08	+5.5264 E-06
$\frac{dC_z}{d\beta_k}$	-2.1190 E-04	9.6692 E-04	-4.6974 E-06	+2.8558 E-04
$\frac{dC_{M_x}}{d\beta_k}$	-3.0110 E-05	+1.2727 E-04	-9.2512 E-07	+5.5924 E-05
$\frac{dC_{M_y}}{d\beta_k}$	+5.8343 E-05	-3.1718 E-04	+1.5573 E-06	-9.7259 E-05
$\frac{dC_{M_z}}{d\beta_k}$	-3.6965 E-06	+9.5445 E-06	-3.0774 E-08	+2.0969 E-06

Table 4.6 (b) Geometric Flap-Deflection SD Ratios ($\frac{\text{Finite Difference}}{\text{QA}}$) Except Terms of $O(\epsilon)$

$\frac{SD_{FD}}{SD_{QA}}$	Design variables $\bar{\beta}$			
	Flap I	Flap II	Flap III	Flap IV
$\frac{dC_x}{d\beta_k}$	0.9999	0.9999	a	a
$\frac{dC_y}{d\beta_k}$	1.0002	1.0001	a	0.9997
$\frac{dC_z}{d\beta_k}$	0.9999	1.0000	0.9998	1.0003
$\frac{dC_{M_x}}{d\beta_k}$	0.9999	1.0000	0.9998	1.0006
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0000	0.9998	1.0006
$\frac{dC_{M_z}}{d\beta_k}$	1.0000	1.0000	a	1.0003

a See note at Table 4.4.

Table 4.6 (c) Geometric Flap-Deflection SD Computational-Time Comparisons

Solution method	Number of solutions	Ratio
Baseline	1	1.000*
Central finite differencing	8	0.877
Quasi-analytical	4	0.3439

* See note at Table 4.3.

Table 4.7 (a) Geometric Planform SD of Force and Moment Coefficients With QAIIM for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

SD_{QA}	Scaled design variables $\bar{\beta}$		
	Root chord	Break chord	Tip chord
$\frac{dC_x}{d\beta_k}$	-1.5421 E-02	+1.0243 E-03	+2.1698 E-05
$\frac{dC_y}{d\beta_k}$	+1.6117 E-01	-5.0936 E-04	+7.1228 E-05
$\frac{dC_z}{d\beta_k}$	+4.7495 E-03	7.7265 E-04	+4.6021 E-05
$\frac{dC_{M_x}}{d\beta_k}$	+7.1231 E-04	+1.1721 E-04	+1.7400 E-05
$\frac{dC_{M_y}}{d\beta_k}$	-7.9255 E-03	-1.9745 E-04	-2.3264 E-05
$\frac{dC_{M_z}}{d\beta_k}$	+2.4522 E-02	-2.9745 E-04	-5.9707 E-05

Table 4.7 (b) Geometric Planform SD Ratios $\left(\frac{\text{Finite Difference}}{QA}\right)$

$\frac{SD_{FD}}{SD_{QA}}$	Scaled design variables $\bar{\beta}$		
	Root chord	Break chord	Tip chord
$\frac{dC_x}{d\beta_k}$	1.0000	1.0000	0.9999
$\frac{dC_y}{d\beta_k}$	1.0000	1.0000	0.9991
$\frac{dC_z}{d\beta_k}$	1.0018	0.9998	0.9999
$\frac{dC_{M_x}}{d\beta_k}$	1.0009	0.9998	1.0001
$\frac{dC_{M_y}}{d\beta_k}$	1.0004	0.9998	0.9997
$\frac{dC_{M_z}}{d\beta_k}$	1.0002	1.0000	1.0005

(Reduction of 3 OM)

Table 4.7 (c) Geometric Planform SD Ratios ($\frac{\text{Finite Difference}}{\text{QA}}$)

$\frac{SD_{FD}}{SD_{QA}}$	Scaled design variables $\bar{\beta}$		
	Root chord	Break chord	Tip chord
$\frac{dC_x}{d\beta_k}$	1.0000	0.9999	0.9999
$\frac{dC_y}{d\beta_k}$	1.0000	0.9999	1.0000
$\frac{dC_z}{d\beta_k}$	0.9999	1.0000	1.0000
$\frac{dC_{M_x}}{d\beta_k}$	0.9999	1.0000	0.9999
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0001	1.0000
$\frac{dC_{M_z}}{d\beta_k}$	1.0000	0.9999	0.9999

(Reduction of 7 OM)

Table 4.7 (d) Geometric Planform SD Ratios ($\frac{\text{Finite Difference}}{\text{QA}}$)

$\frac{SD_{FD}}{SD_{QA}}$	Scaled design variables $\bar{\beta}$		
	Root chord	Break chord	Tip chord
$\frac{dC_x}{d\beta_k}$	0.9999	0.9999	0.9999
$\frac{dC_y}{d\beta_k}$	0.9999	0.9999	1.0000
$\frac{dC_z}{d\beta_k}$	0.9999	1.0000	1.0000
$\frac{dC_{M_x}}{d\beta_k}$	0.9999	1.0000	0.9999
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0001	1.0000
$\frac{dC_{M_z}}{d\beta_k}$	1.0000	0.9999	0.9999

(Reduction of 11 OM)

Table 4.7 (e) Geometric Planform SD Computational-Time Comparisons

Solution method	Number of solutions	Ratio
Baseline	1	1.000*
Central finite differencing	6	1.322
Quasi-analytical (3 OM)	3	0.2046
Quasi-analytical (7 OM)	3	0.2829
Quasi-analytical (11 OM)	3	0.3606

* See note at Table 4.3.

Table 4.8 Force and Moment Coefficients for HSCT 24E at $M_\infty = 2.4$, $\alpha = 0°$, and $\beta = 0°$

C_x ($\approx$ Drag)	0.0044
C_y ($\approx$ Side)	$O(\epsilon)$
C_z ($\approx$ Lift)	–0.0133
C_{M_x} (Roll)	$< O(\epsilon)$
C_{M_y} (Pitch)	0.0055
C_{M_z} (Yaw)	$< O(\epsilon)$

[a] Baseline solution runtime for $\left(\frac{R_{rms}^n}{R_{rms}^1}\right)$ reduction to $\epsilon = 10^{-8}$ on Cray-2 is 827 sec.

Table 4.9 (a) Nongeometric SD of Force and Moment coefficients With QAIIM for HSCT 24E at $M_\infty = 2.4$, $\alpha = 0°$, and $\beta = 0°$

SD_{QA}	Design Variables $\bar{\beta}$		
	M_∞	α	β
$\frac{dC_x}{d\beta_k}$	-0.0024	-0.0225	$O(\epsilon)$
$\frac{dC_y}{d\beta_k}$	$< O(\epsilon)$	$O(\epsilon)$	-0.0614
$\frac{dC_z}{d\beta_k}$	+0.0079	+1.4714	$O(10\epsilon)$
$\frac{dC_{M_x}}{d\beta_k}$	$< O(\epsilon)$	$< O(\epsilon)$	-0.0094
$\frac{dC_{M_y}}{d\beta_k}$	-0.0033	-0.3244	$O(10\epsilon)$
$\frac{dC_{M_z}}{d\beta_k}$	$< O(\epsilon)$	$O(\epsilon)$	-0.0009

72

Table 4.9 (b) Nongeometric SD ratios ($\frac{\text{Finite Difference}}{\text{Quasi-Analytical}}$) except terms of $O(\epsilon)$

$\frac{SD_{FD}}{SD_{QA}}$	Design Variables $\bar{\beta}$		
	M_∞	α	β
$\frac{dC_x}{d\beta_k}$	0.9999	1.0000	a
$\frac{dC_y}{d\beta_k}$	a	a	0.9999
$\frac{dC_z}{d\beta_k}$	0.9999	1.0000	a
$\frac{dC_{M_x}}{d\beta_k}$	a	a	0.9999
$\frac{dC_{M_y}}{d\beta_k}$	0.9999	1.0000	a
$\frac{dC_{M_z}}{d\beta_k}$	a	a	0.9999

[a] Ratio for extremely small quantities is meaningless.

Table 4.9 (c) Nongeometric SD Computational-Time Comparisons

Solutions	Number of solutions	Ratio
Baseline	1	1.000[a]
Central finite differencing	6	3.426
Quasi-analytical	3	0.487

Chapter 5

HSCT AERODYNAMIC OPTIMIZATION STUDIES

The purpose of the initial studies presented in this chapter is simply to indicate the feasibility of using the SD obtained by the IIM in aerodynamic design optimization or MDO procedures. A generic MDO via SA for two disciplines is flowcharted in Fig. 5.1. These initial applications of the 3-D marching Euler code (MARSEN) with efficient geometric SD calculations are for aerodynamic optimization studies in which the CFD and grid-generation codes are considered as separate disciplines. The optimization procedure is demonstrated in the present study for 3-D inviscid, fully supersonic flow over the HSCT 24E configuration.

5.1 Grid Generation and Grid Sensitivity

The geometry processing and grid-generation codes used here [79, 80] take as input the simplified numerical descriptions of configuration components in a wave-drag, or Harris, format. The various component surfaces are first intersected and filleted into a continuous surface; then suitable computational grids are generated. A sample Euler marching grid generated for the HSCT 24E is given in Fig. 5.2. For the present study, geometric SD are propagated from a design-variable parameterization of the HSCT 24E configuration through these surface-processing and volume-grid-generation codes. These latter codes have been linked together, front ended with a 42–variable wing-geometry parameterization [81, 82], and automatically differentiated. The parameterization [81] of the HSCT 24E wing geometry is divided into three variable types: 7 planform variables,

15 section-thickness variables (5 each at the root, break, and tip sections), and 20 camber-surface variables. The geometry parameterization used herein is discussed in appendix D; the camber parameterization used in Ref. [81] has been replaced. As in Ref. [81], propagation of the geometric SD through the automated geometry package is accomplished with the AD [83, 84] precompiler tool ADIFOR (Automated DIfferentiation of FORtran) [9]). Execution of the ADIFOR-enhanced automated geometry package then calculates not only the grid but also the grid SD's with respect to the design variables used in the geometry parameterization. Both are required as input to the flow code, which has been differentiated "by hand".
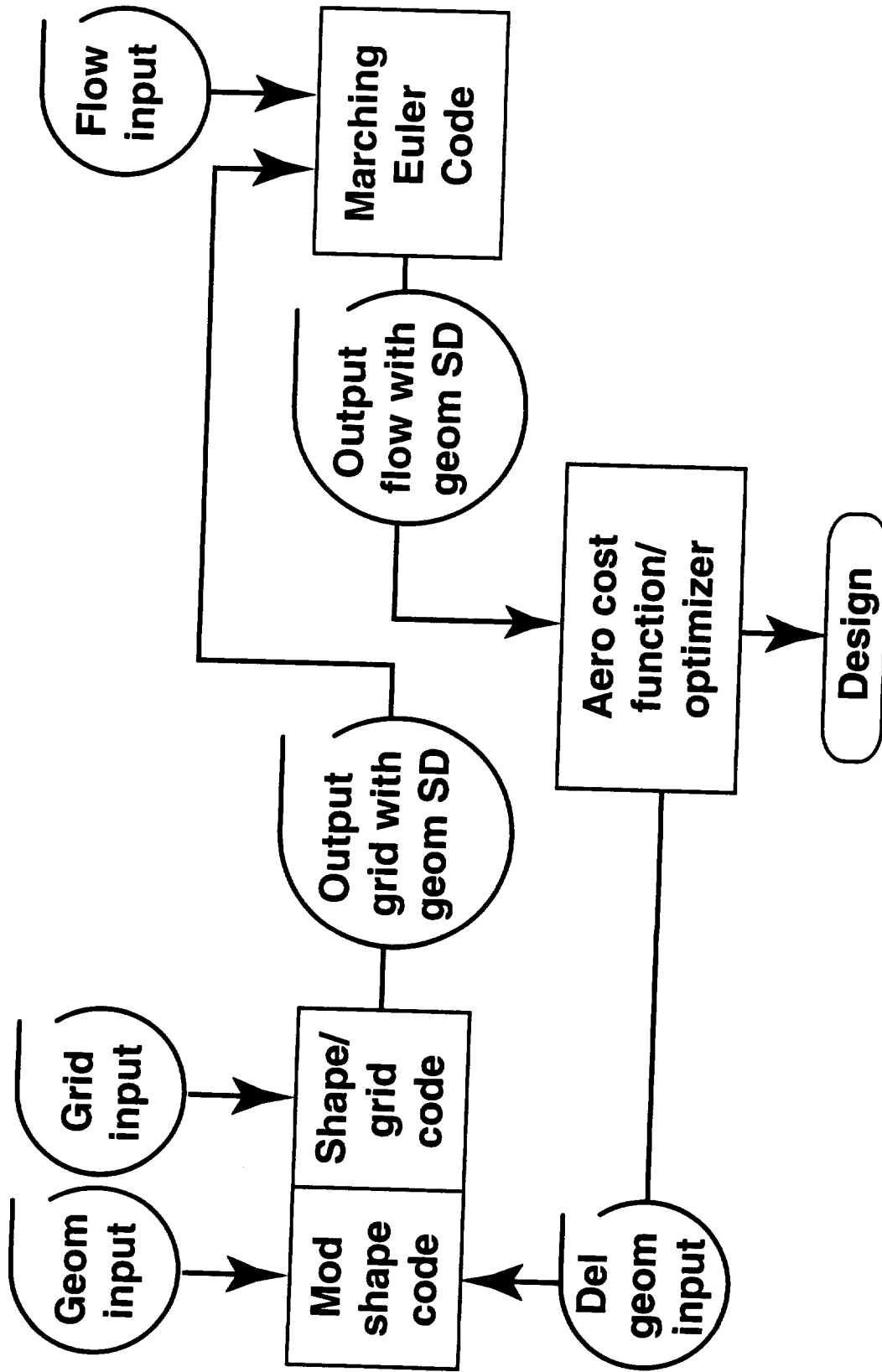
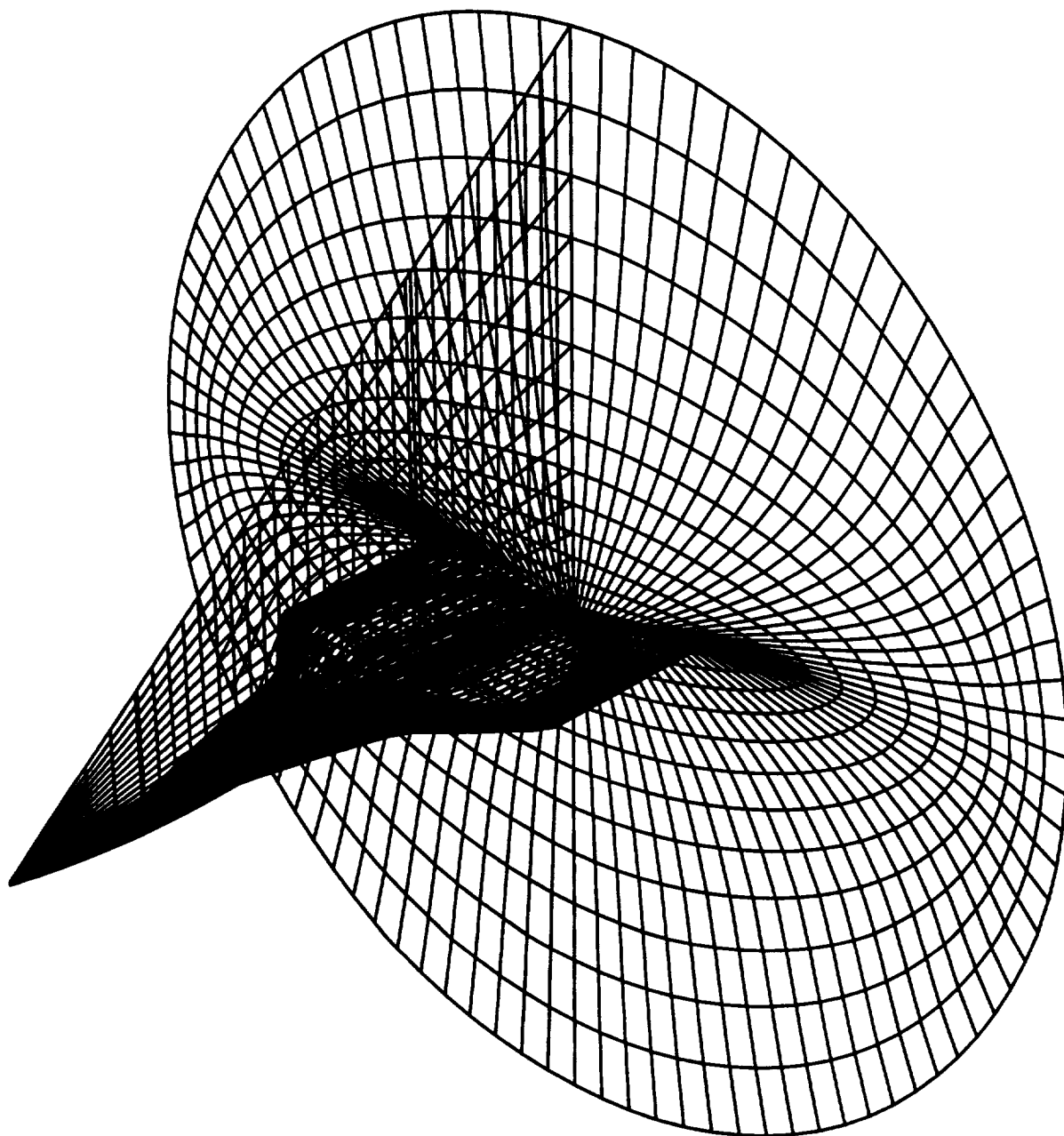Fig. 5.1 Aerodynamic shape optimization: CFD/geometry-grid interaction.

Fig. 5.2 Automated geometry and grid generation for marching Euler code.

5.2 Sample 3–D Optimization Results

The Automated Design Synthesis (ADS) program [85] is used for the optimization code in these studies, basically in a "black box" manner. The two disciplines, CFD and the geometry and grid generation, are coupled sequentially at each optimization step; that is, information passes from the geometry to the grid generation to the flow code with no feedback within each step. The design variables for thickness, camber, flap deflection, and planform have been activated separately to ascertain whether the predicted changes are reasonable when only a supersonic cruise point is considered. The fact that other discipline codes are not participating in the MDO requires that side constraints be specified on the design variables (i.e., with no structural input, minimum thicknesses must be set). Use of the ADS code requires that three options be selected: a strategy, an optimizer, and a one-dimensional search. The following options have been selected for the present constrained optimization results: the sequential quadratic programming strategy, the modified method of feasible directions optimizer, and the Golden section line search. Function and first-order derivative information is given to the ADS code. Because the SD's obtained via the IIM are local derivatives, this combination of methods in ADS appears to provide the most consistent optimization results. However, many function evaluations are required by the selected search procedure.

The HSCT 24E filleted wing-body configuration generated at NASA Langley Research Center is the baseline for these shape-design-improvement studies. These sample studies are done separately for 15 wing-thickness variables, both 28 and 8 wing-camber variables, 4 flap-deflection variables, and 5 wing planform design variables. A summary of results for each of these five studies is given (Tables 5.1 to 5.6, which also will be discussed individually). For these studies, the flow conditions are: $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$ (also noted in each table title). Convergence of both the nonlinear iterative flow analysis and the linear iterative SA was to a relative residual reduction of 6 OM for all required solutions. Extensive use was made of restart solution files for the

flow analysis solutions.

5.2.1 Drag Reduction: Wing-Section Thickness Design Variables

Sample results for the HSCT design improvement study with wing-section thickness variables are given in Table 5.1. Table 5.1(a) is a summary and 5.1(b) gives the initial and final values of the 15 design variables. The 15 thickness design variables are the 5 parameters listed in Table D3 in Appendix D at the wing root, break, and tip locations. The wing thickness is linearly lofted from root to break and break to tip to supply thickness information at all other wing stations (Table D1 in Appendix D). The objective function is drag minimization, and the wing-root bending moment and lift are constrained to their baseline values; that is,

$$\text{minimize} \quad \frac{C_x}{C_{x_0}}$$

$$\text{subject to} \quad \frac{C_{M_x}}{C_{M_{x_0}}} \leq 1.0$$

$$\frac{C_Z}{C_{Z_0}} \geq 1.0$$

The drag improvement evident in Table 5.1(a) is about 10.5 percent, and both constraints are active (within ± 0.5 percent of the baseline value). This improvement was obtained in 8 optimization steps, which required 117 function evaluations and 8 gradient evaluations; the Cray-2 run time was approximately 1.2 hours.

With regard to the run time of the codes on the Cray-2 for a relative residual reduction of 6 OM with 15 design variables, the initial 267 seconds consists of about 67 seconds for an analysis run from a dead start and 200 seconds for the 15 SD evaluations by the IIM. If all function evaluations, including those for the central SD_{FD} required for this study, were done from a dead start (i.e., with a uniform free stream), then the total CPU time would have been about 23,920 seconds or 6.64 Cray-2 hours. Therefore, the total

time savings with the use of restart files is about 18,750 seconds; the savings due to the use of SD evaluations via the IIM is an additional 800 seconds. Note, however, that the time savings due to the use of restart files is code dependent and appears to be large for the present analysis code; the time savings for using SD evaluations via the IIM instead of using SD_{FD} from a dead start would be about 14,480 seconds.

For supersonic flow considerations alone, the wing would be expected to become thinner, which occurs as shown in Fig. 5.3. Table 5.1(b) shows the initial and final values of the 15 thickness design variables and indicates those variables that are influenced by the side constraints (bounds). For 6 of the 15 variables, the side constraints are active (within 5 percent of the specified bounds, which for the thickness variables were taken to be $\pm$ 50 percent of the baseline values). These active side constraints tend to "trap" the optimization in a "corner" of the design parameter space, which may not be realistic because the nonparticipation of the other disciplines has only been mimicked by the side constraints.
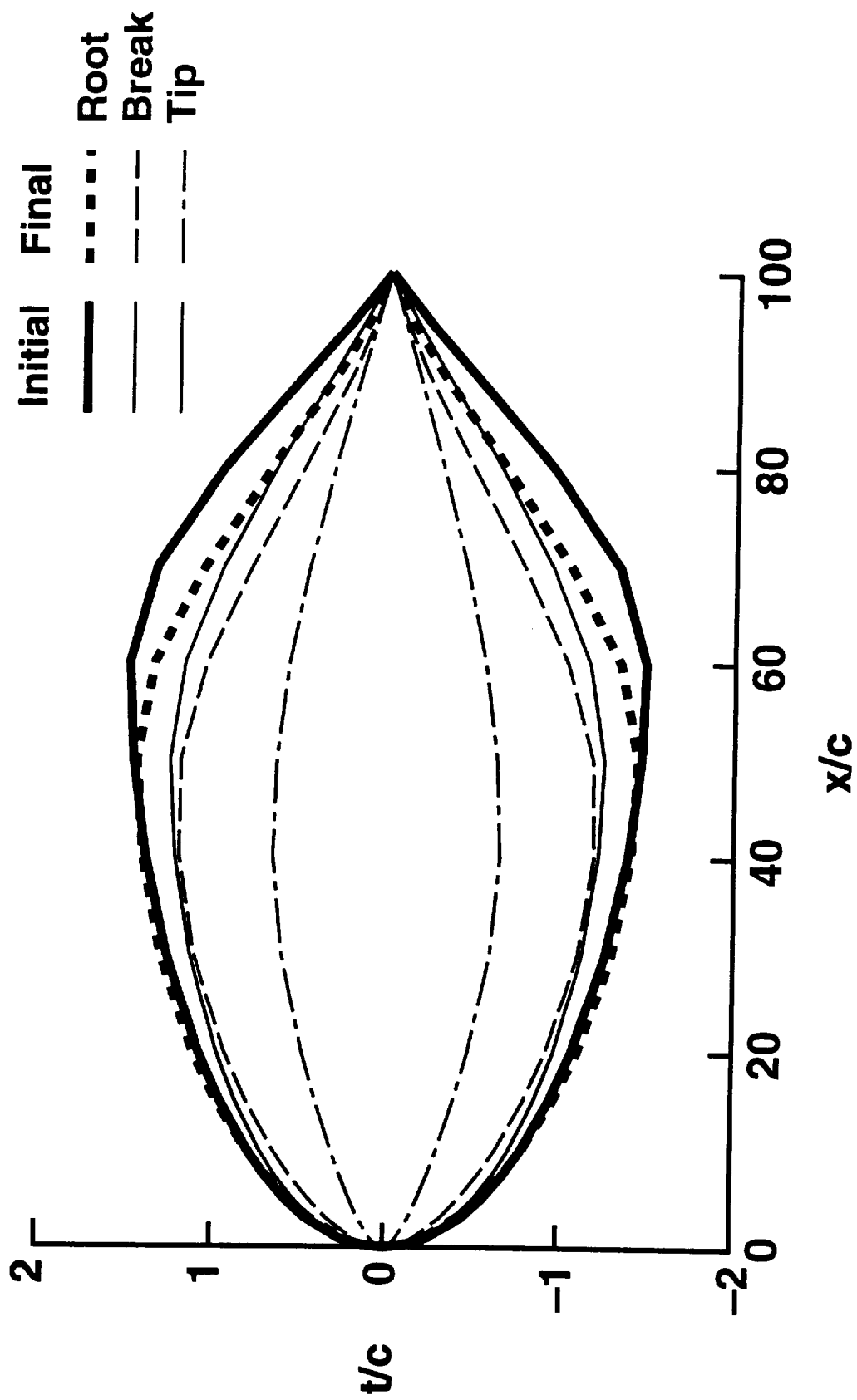
Fig. 5.3 Thickness design improvement
(cruise condition, section thickness distribution).

5.2.2 Lift Improvement: Wing-Section Camber Surface-Elevation Design Variables

Sample results for the HSCT lift-improvement studies with wing-camber surface-elevation design-variables are given in Tables 5.2 and 5.3 for cases with 28 and 8 design variables respectively. In these studies, the camber design variables at the first two wing stations were held constant because the body camber line of the filleted wing-body configuration was fixed and because the wing lofting to determine the body intersection and filleting involved these first two wing stations. The camber surface, for most of the baseline HSCT 24E outboard wing, appeared to vary linearly from just beyond the break to the tip. Therefore, 28 camber variables were active in the first study: 4 each (Table D4 in Appendix D) at wing stations 3 through 8 (break) and at wing station 18 (tip) (Table D1 in Appendix D) with linear lofting from break to tip. Eight camber variables were active in the second study: four each at both wing station 8 (break) and at wing station 18 (tip) with a parabolic lofting from root to break (i.e., a curve that passes through the break variable and the fixed camber variables at wing stations 1 and 2) and with a linear lofting from break to tip. For these studies, the objective and the constraints are

$$\text{minimize} \quad -\frac{C_z}{C_{z_0}}$$

$$\text{subject to} \quad \frac{C_{M_x}}{C_{M_{x_0}}} \leq 1.0$$

$$\frac{C_x}{C_{x_0}} \leq 1.0$$

As shown in Table 5.2(a), a lift improvement of about 7 percent was obtained in nine optimization steps, and the constraints were active. The nine optimization steps required 136 function evaluations and 9 gradient evaluations for 28 design variables. If all function evaluations and central SD_{FD} were done without the restart, the total CPU time would be approximately 42,900 seconds rather than 6680 seconds. The camber

design-variable changes for this improvement study are given in Tables 5.2(b)–(e), for each of the four camber parameters respectively. For 22 of the 28 variables, the side constraints are active.

Contour plots of the camber surface elevation Z_C are compared in Fig. 5.4. The contour plot for the HSCT 24E is shown in Fig. 5.4(a), and the plot from the lift–imrovement study with 28 wing-camber design variables in Fig. 5.4(b). The latter plot appears to be rougher than that for the baseline. The difference is more evident in Fig. 5.5, which compares the spanwise variations of the camber-surface elevations at the wing midchord and the wing trailing edge. As noted in Appendix D, this camber surface elevation includes not only the customary camber parameter A but also a wing-twist parameter ZTE and camber-inflection parameter E. No spanwise control or smoothing was enforced in the 28-variable optimization case.

The purpose of the 8-variable study was to add spanwise control on the adjustment of the wing-section camber design variables. The effect is evidenced in both Fig. 5.4(c) and Fig. 5.5 as a much smoother spanwise variation of the camber surface elevation in comparison with the variation seen in the 28-variable study. Wing lift-improvement results for the 8-variable case are summarized in Table 5.3. The lift increase of approximately 2.6 percent was obtained in eight optimization steps; both constraints, as well as the side constraints on four of the eight design variables, are active. Comments similar to those made about the previously shown sample studies also apply to the CPU times for this case.

(a) - BASELINE HSCT24E WING.
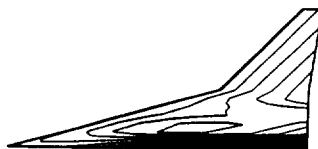
(b) - FINAL HSCT, 28 VARIABLE DESIGN.

(C) - FINAL HSCT, 8 VARIABLE DESIGN.

Fig. 5.4 Camber contours of wing camber surface elevations (contours of constant Z_C) for HSCT lift-improvement studies.
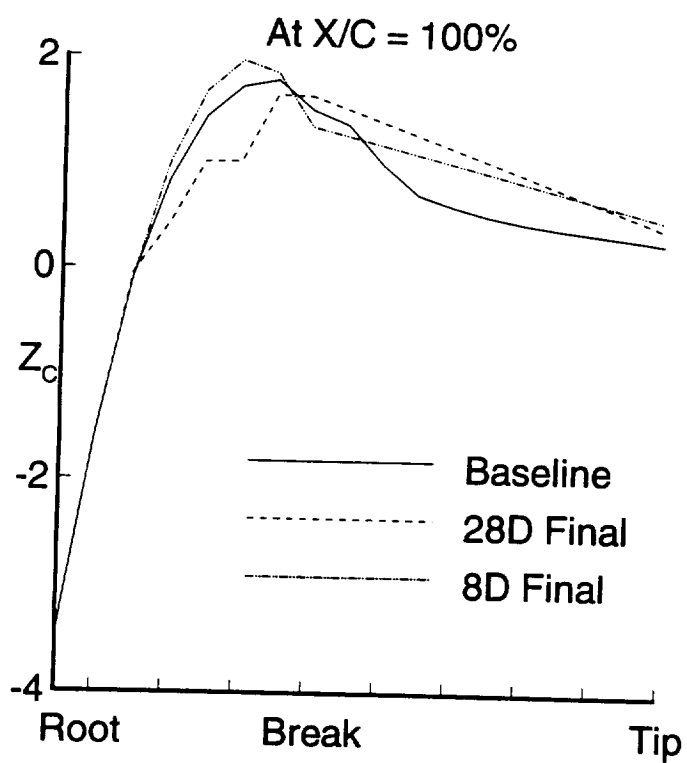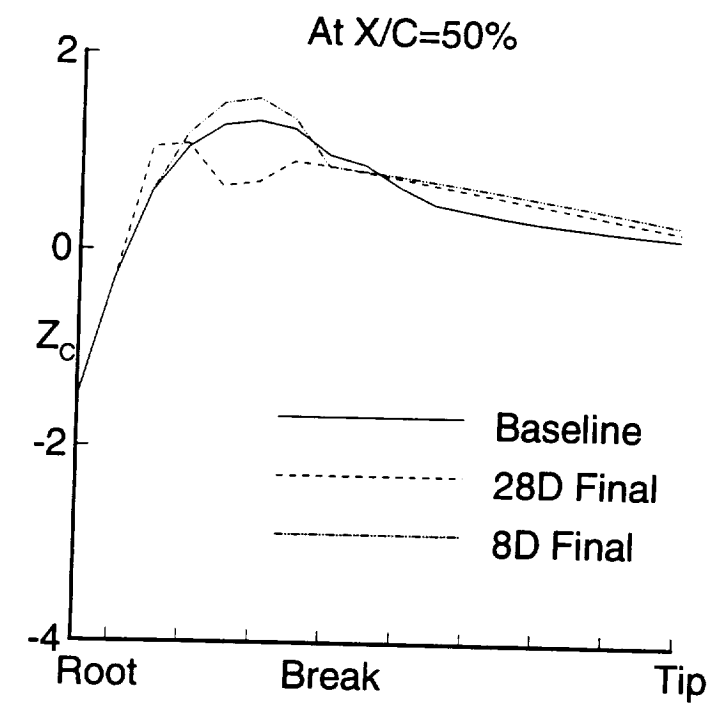
Fig. 5.5 Comparison of spanwise variations of wing camber surface elevation for HSCT lift-improvement study.

5.2.3 Lift Improvement: Flap-Deflection Variables

In MDO applications, all CFD solutions should be provided for at least an approximately deflected and a trimmed configuration. As a first step in this multiple discipline interaction, the static balance and trim control-surface deflections should be investigated for advanced CFD code solutions. Four outboard flaps were defined as part of the baseline HSCT 24E wing; these are shown in Fig. D2 in Appendix D for the design. Typically, the flaps would be "designed" at low-speed flow conditions with takeoff and landing. At high-speed flow conditions, they might be deflected for trim and control purposes. An indication of their effectiveness for lift improvement on the HSCT 24E is demonstrated by the sample results shown in Table 5.4. The objective and constraint functions are the same as in the other lift-improvement studies; here, Table 5.4 shows that a 1-percent lift increase is obtained in five optimization steps and both constraints are active. Initial and final values of the scaled flap deflections are shown in Table 5.4.

Table 5.4 shows that the flap deflection SD's for these outboard flaps are rather small in comparison with the SD's for some of the other geometric design variables. As a result, no attempt has yet been made to trim the pitching moment for the HSCT 24E. Two studies were done, however, on a delta wing for which larger inboard and outboard flaps were defined. In the first study, a lift improvement of 1.2 percent, with bending moment and drag constrained, was obtained in five optimization steps. In the second study, the pitching moment was changed approximately 8.6 percent in six optimization steps, with bending moment, lift, and drag constrained.

5.2.4 Lift Improvement: Wing Planform Design Variables

Planform optimization should be accomplished as a MDO study because input from other disciplines is required. Therefore, planform optimization is typically done (1) early in the design cycle at the conceptual or early preliminary design stages in which these other disciplines participate and (2) with linear aerodynamic codes. Generally, several (or more) discrete planforms are selected, and section variables are then optimized for

each planform study. In the sample case presented in this section, lift optimization for constrained wing bending moment and drag has been done with five planform variables (those shown with solid arrows in Fig. D1 in Appendix D); all other design variables were held at their baseline HSCT 24E values. In the next section, samples of the more conventional camber optimization for different planforms are given and discussed.

Results for lift optimization with respect to five planform variables are given in Table 5.5. A minimum (perhaps a local minimum) has been found in four optimization steps with a lift improvement of 5.5 percent and the drag constraint violated by 3.8 percent. Neither the wing bending-moment constraint nor any of the design-variable side constraints are active or violated.

The baseline and optimized planforms are shown in Fig. 5.6. For supersonic flow considerations alone, the wing tip should be swept more than in the baseline HSCT 24E; Fig. 5.6 shows that the optimization procedure is in agreement with this result. At a Mach number of 2.4, the Mach angle is 24.6°. The angle subtended by the wing-tip leading edge from the root leading edge is 25.9° for the baseline HSCT 24E and 23.8° for the final optimized planform, as depicted in Fig. 5.7. That is, the planform optimized for only supersonic flow lies behind the Mach cone.

Planform optimizations with other objectives (e.g., drag minimization or lift to drag ratio maximization) and different design variables have been completed; however, comprehensive conclusions cannot yet be drawn. In particular, for the optimization results just presented, the planform area changed. In the present study, the geometry and grid-generation codes have not been differentiated with respect to planform area in order to constrain it formally in the optimization. For the double trapezoidal wing planform, this can be done with the three wing chords and two wing spans held fixed, which allows only the inboard and outboard wing panel sweeps to change. (See Fig. D1 in Appendix D)
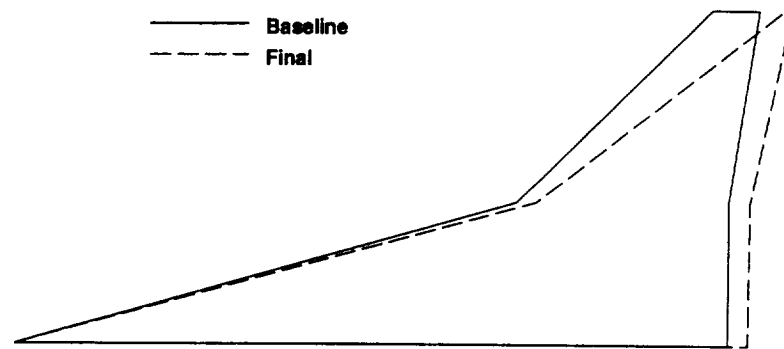
Fig. 5.6 Planform design improvement at cruise condition.
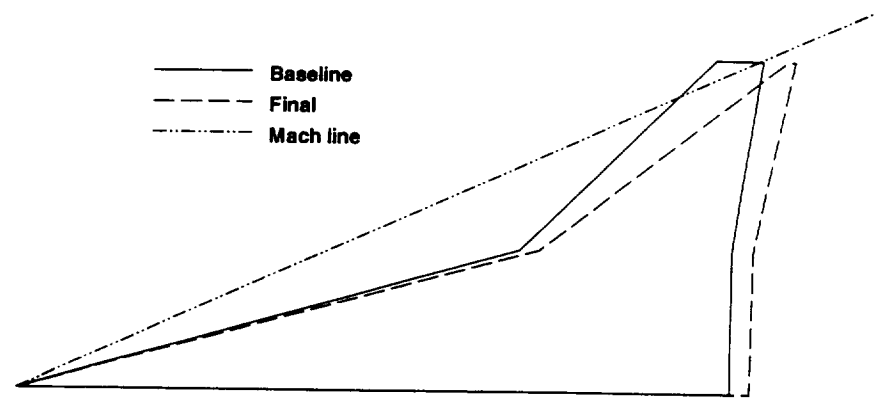
Fig. 5.7 Planform design improvement shown with Mach angle.

5.2.5 Lift Improvement: Camber Variables, Various Planforms

Two planforms that differ from the baseline HSCT 24E were selected for camber optimization studies to improve lift, subject to constrained wing bending moment and drag. The two planforms were a clipped delta wing and a clipped arrow wing with planform area and root chord equal to those for the baseline HSCT 24E. The tip chord for these two clipped planforms was 1/10 of the HSCT 24E tip chord. The leading-edge sweep of the arrow wing was taken to be that of the inboard panel of the HSCT 24E. These three planforms are shown in Fig. 5.8.

A summary of the camber optimization study for the three planforms is given in Table 5.6. The results for the HSCT 24E are those given in Table 5.2 for the 28-variable case; these results have already been discussed in detail. Lift improvement and active constraints occur for all three planforms. The resulting camber surface for the delta wing is rough, as for the HSCT 28-design-variable case previously discussed. The camber surface for the arrow wing was not nearly as rough; however, only three optimization steps were taken. Comments similar to those made previously about the HSCT camber optimization also apply to the CPU times for these two clipped planform studies.

(a) - BASELINE HSCT24E WING
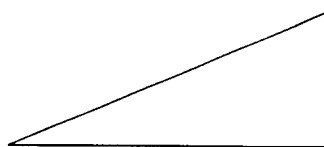
(b) - DELTA WING
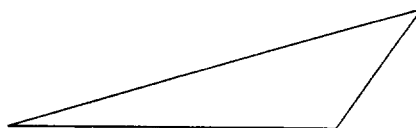
(c) - ARROW WING

Fig. 5.8 Comparison of various planforms for lift-improvement studies.

Table 5.1 (a) Wing Thickness Optimization Study: Design-Improvement Summary with 15 Design Variables for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

	Initial	Final	% Change
Objective (C_x)	1.9361 E-03	1.7311 E-03	−10.59E+00
Constraint I (C_{M_x})	8.4735 E-04	8.4735 E-04	+0.55E-03**
Constraint II (C_z)	1.9086 E-02	1.9087 E-02	+0.68E-02**
Number of function evaluations	1	117	
Number of gradient evaluations	1	8	
CPU time (sec)*	267	4369	

* Run time on Cray-2 for reduction of 6 OM in analysis and SD residuals at every evaluation.

** Active constraint or side constraint on design variable.

Table 5.1 (b) Wing Thickness Optimization Study: Scaled Design-Variable Changes

Design variable	Initial value	Final value	% change
Root I	3.6811	3.2830	−10.8
Break I	4.0288	2.8481	−29.31
Tip I	4.0288	2.1917	-45.60**
Root B	4.8950	5.8788	+20.10
Break B	6.1160	8.6057	+40.71
Tip B	6.1160	8.9049	+45.60**
Root t/C	2.9710	2.8824	−2.98
Break t/C	2.5000	2.4141	−8.59
Tip t/C	2.5000	1.3084	−47.66**
Root Xm	6.0000	5.0874	−15.21
Break Xm	5.0000	4.5458	−9.08
Tip Xm	5.0000	4.1718	−16.56
Root Tau	4.1830	2.1763	−47.97**
Break Tau	2.8980	1.5078	−47.97**
Tip Tau	2.8980	1.5765	−45.60**

Table 5.2 (a) Wing Camber Optimization Study: Design Improvement Summary with 28 Design Variables for HSCT 24E at M_∞ = 2.4, α = 1°, and β = 0°

	Initial	Final	% Change
Objective (C_z)	1.6446 E-02	1.7584 E-02	+6.92
Constraint I (C_{Mx})	4.0315 E-04	4.0228 E-04	−0.22**
Constraint II (C_x)	2.0253 E-03	2.0259 E-03	+0.03**
Number of function evaluations	1	136	
Number of gradient evaluations	1	9	
CPU time (Sec)*	400	6676	

* See note at Table 5.1.
** See note at Table 5.1.

Table 5.2 (b) Wing Camber Optimization Study: Scaled Twist Design-Variable Changes

Wing station	Initial value	Final value	% change
3	9.6	4.820	−49.79**
4	8.22	4.110	−50.00**
5	1.425	0.998	−42.70
6	1.714	0.999	−41.72
7	1.780	1.637	−18.33
8 (break)	1.493	1.624	+8.77
18 (tip)	2.660	3.990	+50.00**

Table 5.2 (c) Wing Camber Optimization Study:
Scaled Camber Design-Variable Changes

Wing station	Initial value	Final value	% change
3	2.780	4.163	+49.75**
4	2.684	3.425	+27.61
5	2.371	1.187	-49.94**
6	1.952	0.976	-50.00**
7	1.508	0.754	-50.00**
8 (break)	1.028	0.514	-50.00**
18 (tip)	1.640	1.977	+20.61

Table 5.2 (d) Wing Camber Optimization Study:
Camber-Inflection Design-Variable Changes

Wing station	Initial value	Final value	% change
3	2.092	1.047	-49.95**
4	1.557	0.780	-49.90**
5	1.228	1.842	+50.00**
6	9.944	4.986	-49.86**
7	7.738	11.607	+50.00**
8 (break)	5.722	8.565	+49.69**
18 (tip)	8.572	12.591	+46.89**

Table 5.2 (e) Wing Camber Optimization Study: Scaled
Maximum-Camber-Location Design-Variable Changes

Wing station	Initial value	Final value	% change
3	4.000	5.994	+49.85**
4	4.000	5.996	+49.90**
5	4.000	2.003	-49.90**
6	4.000	2.000	-50.00**
7	4.000	2.000	-50.00**
8 (break)	4.000	2.000	-50.00**
18 (tip)	5.000	2.500	-50.00**

Table 5.3 (a) Wing Camber Optimization Study: Design-Improvement Summary with 8 Design Variables for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

	Initial	Final	% change
Objective (C_z)	1.5186 E-02	1.5578 E-02	+2.58
Constraint I (C_{M_x})	2.9336 E-04	2.9338 E-04	+0.49 E-02**
Constraint II (C_x)	2.0496 E-03	2.0498 E-03	+0.75 E-02**
Number of function evaluations	1	105	
Number of gradient evaluations	1	8	
CPU time (sec)*	137	2978	

* See note at Table 5.1.
** See note at Table 5.1.

Table 5.3 (b) Wing Camber Optimization Study: Design-Variable Changes

Design variable	Initial value	Final value	% change
Break ZTE	1.4930	1.3318	−10.80
Break A	1.0283	0.9914	−3.59
Break E	5.7222	6.0098	+5.03
Break XMA	4.0000	2.2021	−44.50
Tip ZTE	2.6600	4.7880	+80.00**
Tip A	1.6400	2.9520	+80.00**
Tip E	8.5717	15.4290	+80.00**
Tip XMA	5.000	9.000	+80.00**

** See note at Table 5.1.

Table 5.4 (a) Wing Flap-Deflection Optimization Study: Design-Improvement Summary with 4 Design Variables for HSCT 24E at M_∞ = 2.4, α = 1°, and β = 0°

	Initial	Final	% change
Objective (C_z)	1.9087 E-02	1.9309 E-02	+1.17 E+00
Constraint I (C_{Mx})	8.4736 E-04	8.4727 E-04	–0.10 E-02**
Constraint II (C_x)	1.9361 E-03	1.9361 E-03	+0.63 E-05**
Number of function evaluations	1	76	
Number of gradient evaluations	1	5	
CPU time (sec)*	99	1581	

* See note at Table 5.1.
** See note at Table 5.1.

Table 5.4 (b) Wing Flap Deflection Optimization Study: Scaled Design-Variable Changes

Flap number	Initial value	Final value
I	0	–2.4125
II	0	+ 0.2644
III	0	+10.000
IV	0	–1.7263

Table 5.5 (a) Wing Planform Optimization Study: Design-Improvement Summary with 5 Design Variables for HSCT 24E at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

	Initial	Final	% change
Objective (C_z)	1.9086 E-02	2.0133 E-02	+5.5
Constraint I (C_{M_x})	8.4736 E-04	8.4153 E-04	−0.69
Constraint II (C_x)	1.9361 E-03	2.0104 E-03	+3.83***
Number of function evaluations	1	102	
Number of gradient evaluations	1	4	
CPU time (sec)*	132	2701	

* See note at Table 5.1.
*** Constraint violated.

Table 5.5 (b) Wing Planform Optimization Study: Scaled Design-Variable Changes

Design Variable	Initial value	Final value	% change
Root chord	1.420	1.456	+2.52
Break chord	4.236	4.269	+3.24
Tip chord	9.303	1.488	−84.00
X break Leading Edge	9.965	10.358	+3.94
X tip Leading Edge	13.840	15.263	+10.28

Table 5.6 Wing Camber Optimization Study: Summary
for Various Planforms at $M_\infty = 2.4$, $\alpha = 1°$, and $\beta = 0°$

	HSCT	DELTA	ARROW
Objective (C_z), %	+ 6.92	+5.17	+3.23
Constraint I (C_{Mx}), %	–0.22 E+00**	–0.52 E-04**	–0.68 E+00
Constraint II (C_x), %	+0.32 E-01**	+0.81 E-01**	+0.43 E+00**
Number of function evaluations	136	150	53
Number of gradient evaluations	9	9	3
CPU time (sec)*	6676	6888	2760

* See note at Table 5.1.
** See note at Table 5.1.

Chapter 6

SUMMARY AND CONCLUSIONS

The Incremental Iterative Method (IIM) is developed to calculate consistent, discrete sensitivity derivatives (SD's). The method is successfully implemented in the calculation of consistent, discrete SD's for the two-dimensional (2-D) thin layer Navier-Stokes equations and the three-dimensional (3-D) Euler equations. The lift-corrected far-field boundary condition is implemented in the 2-D aerodynamic analysis code and sensitivity analysis (SA) code.

The SD's obtained in two dimensions with the direct-differentiation and adjoint-variable approaches are compared with SD's from finite differences for accuracy and efficiency. Not only do the results from these two methods compare well with those from the finite-difference approach, they are computationally less expensive to obtain. In two dimensions, these methods are applied to two example airfoil problems: subsonic low-Reynolds-number laminar flow and transonic high-Reynolds-number turbulent flow, for which the three geometric design variables and three nongeometric design variables (Mach number, angle of attack, and Reynolds number) are considered. The SD's obtained for the turbulent flow case do not agree "exactly" with the finite-difference results, as expected, because the differentiation of the turbulence terms is neglected due to the complexity of these terms; for the most part, this error was small, but in a few cases, it was significant.

The SD's obtained in three dimensions with the direct-differentiation approach are compared with finite differences for accuracy and efficiency. In three dimensions, this

procedure is demonstrated on the High-Speed Civil Transport (HSCT) configuration generated at NASA Langley Research Center, and SD's are obtained with respect to three nongeometric design variables (Mach number, angle of attack, and yaw angle) and many geometric design variables.

After successful implementation of the IIM in two and three dimensions, these SD's are used in a gradient-based design optimization. Planform, thickness, and camber design improvement studies are done for the HSCT 24E for a supersonic cruise condition with efficiently calculated SD's via the IIM. Remarks in regard to the design-improvement study are summarized as follows:

1. Formulation of the optimization problem is critical. Based on how a problem is posed, the optimization procedure may give completely different answers.

2. An optimization procedure that uses local exact derivatives should not take large step sizes in the design variables.

3. A certain degree of robustness is required in all steps of the optimization. For example, in the present study, the surface/volume-grid generation procedure failed to generate the grid for certain shapes generated by the optimizer.

This IIM is very general and can be easily implemented in any existing CFD code to obtain SD's. Approximations of convenience can be introduced in the matrix operator and thus the same solver that is used for aerodynamic analysis can be used for the SA. Tools like ADIFOR can be used to construct the right-hand side of the sensitivity equation in incremental iterative form. This method currently is being implemented in TLNS3D, for example to calculate SD's. Furthermore, efforts are underway at Argonne National Laboratories to construct a template that can differentiate any CFD code with the IIM.

The design-package code developed in this study can be used for static balance and trim control of the HSCT 24E configuration, in which the objective is to stabilize the configuration with flap deflection as the design variable. This design-package code

can also be coupled with a finite-element structures code for aeroelastic studies and for multidisciplinary design optimization studies in which structures and aerodynamics are treated as separate disciplines: this effort is currently under investigation. The marching Euler code, equipped with the capability to calculate efficient SD's can also be used for shock-wave propagation and sonic boom studies for the HSCT 24E configuration. The single-block marching Euler code developed in this study can be extended to a multiblock version with the added capability to perform viscous calculations. The viscous terms can be differentiated with ADIFOR, and the resulting differentiated code can be coupled with the existing hand-differentiated code, MARSEN (marching Euler sensitivities), to calculate the SD's.

Currently, the linearized system for aerodynamic analysis and the linear system for SA are solved with the spatially split approximate factorization algorithm. To further improve efficiency in solving the linear system, a Krylov-subspace-based method, such as the Generalized Minimal Residual (GMRES) solver, can be added to the existing code. As an additional future application, the IIM can be used on unstructured grids to calculate SD's: unstructured grids can be used more easily than structured grids to model complicated geometries such as the HSCT 24E.

REFERENCES

1. Thomas, J. L., Taylor, S. L., and Anderson, W. K., "Navier-Stokes Computations of Vortical Flows Over Low Aspect Ratio Wings," AIAA Paper 87–0207.

2. Vatsa, V. N., and Wedan, B. W., "Development of a Multigrid Code For 3–D Navier-Stokes Equations and Its Application to a Grid Refinement Study," *Journal of Computers and Fluids*, Vol. 18, No. 4, 1990, pp. 391–403.

3. Jameson, A., "Successes and Challenges in Computational Aerodynamics," AIAA Paper No. 87–1184–CP, June 1987.

4. Coen, P. G., Sobieski, J. S-., Dollyhigh, S., "Preliminary Results from the High-Speed Airframe Integrated Research Project," AIAA Paper No. 92–1004, February 1992.

5. Holst, T. L., Salas, M. D., and Claus, R. W., "The NASA Computational Aerosciences Program-Toward Teraflops Computing," AIAA Paper No. 92–0558, January 1992.

6. Sobieski, J. S-., "Multidisciplinary Optimization for Engineering Systems: Achievements and Potential," NASA TM 101566, March 1989.

7. Sobieski, J. S-., "The Case for Aerodynamic Sensitivity Analysis," NASA CP-2457, January 1987, pp. 77–96.

8. MACSYMA Reference Manual, Version 13, Computer Aided Mathematics Group, Symbolics, Inc., 1988.

9. Bischof, C. H., Carle, A., Corliss, G. F., Griewank, A., and Hovland, P, "ADIFOR: Generating Derivative Codes from Fortran Programs," *Journal of Scientific Programming*, Vol. 1, No. 1, 1992, pp. 1–29.

10. Hutchison, M. G., Unger, E. R., Mason, W. H., Grossman, B., and Haftka, R. T., "Variable-Complexity Aerodynamic Optimization of High-Speed Civil Transport wing," *Journal of Aircraft*, Vol. 31, No. 1, January-February 1994, pp. 110–116.

11. Gill, P. E., Murray, W., Saunders, M. A., and Wright, M. H., "Computing the Finite-Difference Approximations to Derivatives for Numerical Optimization," SOL 80–6 (Contract 29–79–C-0110), Dep. Operations Research-SOL, Stanford University, May 1980.

12. Iott, J., Haftka, R. T., and Adelman, H. M., "Selecting Step Sizes in Sensitivity Analysis by Finite Differences," NASA TM 86382, 1985.

13. Taylor, A. C., Newman, P. A., Hou, G. J.-W., and Jones, H. E., "Recent Advances in Steady Compressible Aerodynamic Sensitivity Analysis," Volumes in Mathematics and its Applications (IMA), Vol. 68, 1994, pp. 341–356; Volume title: *Flow Control*, edited by Professor Max D. Gunzburger, published by Springer-Verlag.

14. Pironneau, O., *Optimal Shape Design for Elliptic Systems*, Springer, NY, 1985.

15. Angrand, F., "Optimum Design for Potential Flows," *International Journal of Numerical Methods in Fluids*, Vol. 3, 1983, pp. 265–282.

16. Yates, E.C., Jr., "Aerodynamic Sensitivities from Subsonic, Sonic, and Supersonic Unsteady, Nonplanar Lifting-Surface Theory," NASA TM-100502, September 1987.

17. Yates, E.C., Jr., and Desmarais, R., "Boundary Integral Method for Calculating Aerodynamic Sensitivities with Illustration for Lifting Surface Theory," in Proceedings of the International Symposium of Boundary Element Methods (IBEM 89), published by Springer-Verlag, October 2-4, 1989, East Hartford, Conn.

18. Jameson, A., "Aerodynamic Design Via Control Theory," *Journal of Scientific Computing*, Vol. 3, 1988, pp. 233–260. (Also NASA CR-181749 and ICASE Report No. 88–64, November 1988.)

19. Jameson, A., "Automatic Design of Transonic Airfoils to Reduce Induced Pressure Drag," Princeton University MAE Report 1881, 1990. (Also 31st Israel Annual Conference in Aviation and Aeronautics, February 1990).

20. Jameson, A., and Reuther, J., "Control Theory Based Airfoil Design Using Euler Equations", Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, 1994, pp. 206–222 (AIAA Paper 94–4272–CP).

21. Frank P.D., and Shubin, G.R., "A Comparison of Optimization-Based Approaches for a Model Computational Aerodynamics Design Problem," *Journal of Computational Physics*, Vol. 98, No. 1, January 1992, pp. 74–89.

22. Shubin, G.R., and Frank, P.D. "A Comparison of the Implicit Gradient Approach and the Variational Approach to Aerodynamic Design Optimization," Applied Mathematics and Statistics Technical Report, AMS-TR-163, Boeing Computer Services, Seattle, Washington, April 1991.

23. Shubin, G.R., "Obtaining 'Cheap' Optimization Gradients from Computational Aerodynamics Codes," Applied Mathematics and Statistics Technical Report, AMS-TR-164, Boeing Computer Services, Seattle, Washington, June 1991.

24. Borgaard, J., and Burns, J., "A Sensitivity Equation Approach to Optimal Design of Nozzles," Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, 1994, pp. 232–241. (AIAA Paper 94-4274-CP).

25. Borgaard, J., Burns, J. A., Cliff, E., and Gunzburger, M., "Sensitivity Calculations For a 2-D Inviscid, Supersonic Forebody Problem," NASA CR-19144 and ICASE Report No. 93–13, March 1993.

26. Ibrahim, A. H., and Baysal, O., "Design Optimization Using Variational Methods and CFD", 32nd Aerospace Sciences Meeting, January 10–13, 1994, Reno, NV, AIAA Paper 94–0093.

27. Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients Based on the Transonic Small Perturbation Formulation," *Journal of Aircraft*, Vol. 27, No. 6, June 1990, pp. 507-515 (also AIAA Paper 89–0532).

28. Elbanna, H. M., and Carlson, L. A., "Determination of Aerodynamic Sensitivity Coefficients Based on the Three-Dimensional Full Potential Equation," AIAA Paper 92-2670, June 1992.

29. Baysal, O., and Eleshaky, M. E., "Aerodynamic Sensitivity Analysis Methods for the Compressible Euler Equations," *ASME Journal of Fluids Engineering*, Vol. 113, No. 4, December 1991. (Also in *Recent Advances and Applications in CFD* , ed. by O. Baysal, ASME-FED Vol. 103, 11th Winter Annual Meeting, November, 1990, pp. 191–202).

30. Baysal, O., Eleshaky, M. E., and Burgreen, G.W., "Aerodynamic Shape Optimization Using Sensitivity Analysis on Third-Order Euler Equations," Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991, Honolulu, Hawaii, AIAA Paper 91-1577.

31. Burgreen, G. W., Baysal, O., and Eleshaky, M. E., "Improving the Efficiency of Aerodynamic Shape Optimization Procedures," in AIAA CP-9213, Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, September 1992, pp. 87–97, (AIAA Paper 92-4697-CP).

32. Eleshaky, M. E., and Baysal, O., "Airfoil Shape Optimization Using Sensitivity Analysis on Viscous Flow Equations," *ASME Journal of Fluids Engineering*, Vol. 115, No.1, March 1993, pp. 75–84. (Also in Multidisciplinary Applications Of

Computational Fluid Dynamics, ed. O. Baysal, ASME–FED, Vol. 129, 12th Winter Annual Meeting, December 1991, pp. 27–37).

33. Taylor, A. C. III, Korivi, V. M., and Hou, G. -W., "Taylor Series Approximation of Geometric Shape Variation For the Euler Equations," *AIAA Journal*, Vol. 30, No.8, August 1992, pp. 2163–2165. (also AIAA 91–0173, January 1991).

34. Taylor, A. C. III, Hou, G. -W., and Korivi, V. M., " Methodology for Calculating Aerodynamic Sensitivity Derivatives," *AIAA Journal*, Vol. 30, No. 10, October 1992, pp. 2411–2419. (Also in Proceedings of the AIAA/ASME/ASCE/AHS/ASC 32nd Structures, Structural Dynamics, and Materials Conference, April 1991, pp. 477–489, AIAA Paper 91-1101–CP).

35. Hou, G. -W., Taylor, A. C. III, and Korivi, V. M., "Discrete Shape Sensitivity Equations For Aerodynamic Problems," *International Journal For Numerical Methods in Engineering*, Vol. 37, 1994, pp. 2251–2266. (Also AIAA Paper 91-2259, June 1991.)

36. Korivi, V.M., "Sensitivity Analysis Applied to the Euler Equations," M.S. Thesis, Old Dominion University, Norfolk, VA, August, 1991.

37. Taylor, A.C., III, Korivi, V.M., and Hou, G.W. "Approximate Analysis and Sensitivity Analysis Methods For Viscous Flow Involving Variation of Geometric Shape," Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, June 24-26, 1991, Honolulu, Hawaii, AIAA Paper 91-1569.

38. Taylor, A.C. III, Hou, G.W., and Korivi, V.M., "An Efficient Method For Estimating Steady-State Numerical Solutions to the Euler Equations," AIAA Paper 91–1680, June 1991.

39. Taylor, A.C. III, Hou, G.W., and Korivi, V.M., "Sensitivity Analysis, Approximate Analysis, and Design Optimization For Internal and External Viscous Flows," AIAA Paper 91–3083, September 1991.

40. Eleshaky, M. E., and Baysal, O., "Preconditioned Domain Decomposition Scheme for Three-Dimensional Aerodynamics Sensitivity Analysis," in AIAA CP-933, AIAA 11th Computational Fluid Dynamic Conference, July 1993, pp. 1055–1056.

41. Lacasse, J. M., and Baysal, O., "Shape Optimization of Single and Two Element Airfoils on Multiblock Grids", Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, 1994, pp. 223–231 (AIAA Paper 94–4273–CP).

42. Korivi, V. M., Taylor, A. C. III, Newman, P. A., Hou, G. J.-W., and Jones, H. E., "An Approximately-Factored Incremental Strategy for Calculating Consistent

Discrete Aerodynamic Sensitivity Derivatives," *Journal of Computational Physics*, Vol. 113, No. 2, August 1994, pp. 336–346. (AIAA Paper 92–4746–CP, September 1992).

43. Newman, P. A., Hou, G. J.-W., Jones, H. E., Taylor, A. C. III, and Korivi, V. M., "Observations on Computational Methodologies for Use in Large-Scale Gradient-Based Multidisciplinary Design," Proceedings of the Fourth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Cleveland, OH, 1992, pp. 531–542 (AIAA Paper 92–4753–CP, September 1992).

44. Korivi, V. M., Taylor, A. C. III, Hou, G. J.-W., Newman, P. A., and Jones, H. E., "Sensitivity Derivatives for Three-Dimensional Supersonic Euler Code using Incremental Iterative Strategy," A Collection of Technical Papers, Part 2, Proceedings of the 11th AIAA Computational Fluid Dynamics Conference, AIAA, Orlando, FL, 1993, pp. 1053–1054 (expanded in *AIAA Journal*, Vol. 32, No. 6, June 1994, pp 1319–1321).

45. Chattopadhya, A., and Pagaldipti, N., "A Multilevel Decomposition Procedure for Efficient Design of High Speed Civil Transport," 32nd Aerospace Sciences Meeting & Exhibit, January 10–13, 1994, Reno, NV, AIAA Paper 94–0097.

46. Pagaldipti, N., and Chattopadhya, A., "A Discrete Semi-Analytical Procedure for Aerodynamic Sensitivity Analysis Including Grid Sensitivity", Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, 1994, pp. 161–169 (AIAA Paper 94–4268–CP).

47. Huddleston, D. H., Soni, B. K., and Zheng, X., "Application of a Factored Newton-Relaxation Scheme to Calculation of Discrete Aerodynamic Sensitivity Derivatives," 12th AIAA Applied Aerodynamics Conference, June 1994, Colorado Springs, CO, AIAA Paper 94–1894.

48. Dulikravich, G. S., "Aerodynamic Shape Design and Optimization: Status and Trends," *Journal of Aircraft*, Vol. 29, No. 6, November-December 1992, pp. 1020–1026.

49. Rizk, M., "The Single-Cycle Scheme: A New Approach to Numerical Optimizations," *AIAA Journal*, Vol. 21, 1983, pp. 1640–1647.

50. Rizk, M., "Optimization by Updating Design Parameters as CFD Iterative Flow Solutions Evolve," in Multidisciplinary Applications of Computational Fluid Dynamics, ed. by O. Baysal, ASME-FED, Vol. 129, Winter Annual Meeting, December 1991, pp. 51–62.

51. Ghattas, O., and Xiaogang, Li "A Variational Finite Element Method for Non-linear Fluid-Solid Interaction and its Sensitivity Analysis," AIAA Paper 94-4399, September 1994.

52. Hou, G. J. -W., Taylor, A. C., Mani, S. V., and Newman, P. A., "Simultaneous Aerodynamic Analysis and Design Optimization," abstract in Second U.S. National Congress on Computational Mechanics, August 16–18, 1993, Washington, D.C.

53. Ta'asan, S., Kuruvila, G., and Salas, M. D., "Aerodynamic Design and Optimization in One Shot," AIAA Paper 92-0025, January 1992.

54. Kuruvila, G., Ta'san, S., and Salas, M. D., " Airfoil Optimization by the One-Shot Method," in AGARD-FDP-VKI Special Course on Optimum Design Methods in Aerodynamics, Rhode-Saint-Genese, Belgium, 25–29 April 1994.

55. Huffman, W. P., Melvin, R. G., Young, D. P., Johnson, F. T., Bussoletti, J. E., Bieterman, M. B., and Hilmes, C. L., "Practical Design and Optimization in Computational Fluid Dynamics," AIAA Paper 93-3111, July 1993.

56. Burgreen, G. W., "Three-Dimensional Aerodynamic Shape Optimization of Wings Using Discrete Sensitivity Analysis", Ph.D Dissertation, Old Dominion University, Norfolk, Virginia, May 1994.

57. Burgreen, G. W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Wings Using Sensitivity Anlysis," 32nd Aerospace Sciences Meeting&Exhibit, January 10–13, 1994, Reno, NV, AIAA Paper 94-0094.

58. Burgreen, G. W., and Baysal, O., "Three-Dimensional Aerodynamic Shape Optimization of Supersonic Delta Wings", Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, 1994, pp. 195–205 (AIAA Paper 94-4271-CP).

59. Jameson, A., "Optimum Aerodynamic Design Via Boundary Control", in AGARD-FDP-VKI Special Course on Optimum Design Methods in Aerodynamics, April 25–29, 1994, Von Karman Institute, Rhode-St Genese, Belgium.

60. Korivi, V. M., Newman, P. A., and Taylor, A. C., "Aerodynamic Optimization Studies Using a 3-D Supersonic Euler Code with Efficient Calculation of Sensitivity Analysis", Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, 1994, pp. 170–194 (AIAA Paper 94-4270-CP).

61. Thomas, J. L, and Salas M. D., "Far-Field Boundary Conditions for Transonic Lifting Solutions to the Euler Equations," AIAA Journal, Vol. 24, No. 7, July 1986, pp.

1074-1080 (also AIAA Paper 85-0020).

62. Walters, R. W., and Thomas, J. L., "Advances in Upwind Relaxation Methods," in State of the Art Surveys of Computational Mechanics, ed. A.K. Noor, pp. 145-183, ASME Publication, 1989, New York.

63. Thomas, J.L., Van Leer, B., and Walters, R.W., "Implicit Flux-Split Schemes for the Euler Equations," *AIAA Journal*, Vol. 28, No. 6, June 1990, pp. 973-974 (also AIAA Paper 85-1680).

64. Van Leer, B., "Flux-Vector Splitting for the Euler Equations," ICASE Report 82-30, September 1982 (also *Lecture Notes in Physics*, Vol. 170, 1982, pp. 507-512).

65. Taylor, A. C., "Convergence Acceleration of Upwind Relaxation Methods for the Navier-Stokes equations," Ph.D. Dissertation, Virginia Polytechnic Institute and State University, July 1989.

66. Beam, R.M. and Warming, R.F., "An Implicit Factored Scheme for the Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 16, April 1978, pp. 393–402.

67. Yoon, S., and Jameson, A., "An LU-SSOR Scheme for the Euler and Navier-Stokes Equations," AIAA Paper 87–0600, January 1987.

68. Thomas, J.L., and Walters, R.W., "Upwind Relaxation Algorithms for the Navier-Stokes Equations," *AIAA Journal*, Vol. 25, No. 4, April 1987, pp. 527–534.

69. Walters, R.W., Dwoyer, D.L., and Hassan, H.A., "A Strongly Implicit Procedure for the Compressible Navier Stokes Equations," *AIAA Journal*, Vol. 24, No. 1, January 1986, pp. 6-12.

70. Ajmani, K., Ng, W.F., and Liou, M.S., "Generalized Conjugate-Gradient Methods for the Navier-Stokes Equations," Proceedings of the AIAA 10th Computational Fluid Dynamics Conference, June 24–26, 1991 Honululu, Hawaii, AIAA Paper 91–1556.

71. Venkatakrishnan, V., "Preconditioned Conjugate Gradient Method For The Compressible Navier-Stokes Equations," *AIAA Journal*, Vol. 29, No. 7, July 1991, pp. 1092-1100 (also AIAA Paper 90–0586).

72. Newsome, R. W., Walters, R. W., and Thomas, J. L., "An Efficient Strategy for Upwind/Relaxation Solutions to the Thin-Layer Navier-Stokes Equations," AIAA Paper 87–113.

73. Smith, R.E., Jr., and Sadrehaghighi, I., "Grid Sensitivity in Airplane Design," in Proceedings of the 4th International Symposium of Computational Fluid Dynamics, September 9–12 1991, University of California-Davis, pp. 1071–1077.

74. Sadrehaghighi, I., Smith, R.E., Jr., and Tiwari, S.N., "An Analytical Approach To Grid Sensitivity Analysis," AIAA Paper 92–0660, January 1992.

75. Rajan, S. D., and Belegundu, A. D., "A Shape Optimization Approach using Fictitious Loads as Design Variables," Proceedings of the AIAA/ASME/ASCE/AHS 28th Structures, Structural Dyanamics, and Material Conference, April 6–8, 1987, Monterry CA, also AIAA Paper 87–0834.

76. Choi, K. K. and Yao, T. M., "3–D Modelling and Automatic Regridding in Shape Design Sensitivity Analysis," in *Sensitivity Analysis in Engineering*, NASA Conference Publication 2457, September 1986, pp. 329–346.

77. Green, L. L., Bischof, C. H., Carle, A., Griewank, A., Haigler, K., and Newman, P. A., "Automatic Differentiation of Advanced CFD Codes With Respect To Wing Geometry Parameters For MDO," abstract in Second U.S. National Congress on Computational Mechanics, August 16–18 1993, Washington, D.C.

78. Baldwin, B., and Lomax, H., "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78–0257, January 1978.

79. Barger, R. L., and Adams, M. S., "Automatic Computation of Wing-Fuselage Intersection Lines And Fillet Inserts With Fixed-Area Constraint," NASA TM 4406, March 1993.

80. Barger, R. L., Adams, M. S., and Krishnan, R. R., "Automatic Computation of Euler Marching Grids and Subsonic Grids for Wing-Fuselage Configurations," NASA TM 4573, July 1994.

81. Unger, E., and Hall, L. E., "The Use of Automatic Differentiation in an Aircraft Design Problem," Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, Panama City, FL, September 1994, pp. 64–72. (AIAA Paper 94–4260–CP).

82. Unger, E., Private Communication, 1993.

83. Rall, L. B., "Automatic Differentiation: Techniques and Applications," Volume 120 of *Lecture Notes in Computer Science, Mathematical Programming: Recent Developments and Applications*, Springler Verlag, Berlin, Germany, 1981.

84. Griewank, A., and Corliss. G. F., eds: *Automatic Differentiation of Algorithms: Theory, Implementation, and Application*, SIAM, Philadelphia, PA, 1991.

85. Vanderplaats, G. N., "ADS- A Fortran Program for Automated Design Synthesis," NASA CR-17785, NASA Ames Research Center, September 1985.

86. Bischof, C. H., and Griewank, A., "ADIFOR: A Fortran System for Portable Automatic Differentiation," Proceedings of the AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization", AIAA Cleveland, OH, September 1992, pp. 433–441 (AIAA 92–4744–CP, September 1992).

87. Bischof, C. H., Carle, A., Corliss, G. F., Griewank, A., and Hovland, P, "Getting Started With ADIFOR. ADIFOR Working Note #9," ANL-MCS-TM-164, Mathematics and Computer Science Division, Argonne National Laboratory, 1992.

88. Bischof, C. H., and Hovland, P., "Using ADIFOR to compute Dense and Sparse Jacobians. ADIFOR Working Note #2," ANL-MCS-TM-158, Mathematics and Computer Science Division, Argonne National Laboratory, 1991.

89. Bischof, C. H., Corliss, G., Green, L. L., Griewank, A., Haigler, K., and Newman, P. A., "Automatic Differentiation of Advanced CFD Codes for Multidisciplinary Design," presented at the Symposium on High-Performance Computing For Flight Vehicles, December 1992, Arlington, VA (to appear in *Computing Systems in Engineering*, Vol. 3, No. 6, 1993).

90. Green, L. L., Newman, P. A., and Haigler, K. J., "Sensitivity Derivatives For Advanced CFD Algorithm and Viscous Modelling Parameters Via Automatic Differentiation," in AIAA CP-277 AIAA 11th Computational Fluid Dynamics Conference, July 1993, pp. 260–277. (AIAA Paper 93–3321–CP)

91. Sherman, L. L., Taylor, A. C., Green, L. L., Newman, P. A., Hou, G. J.-W., and Korivi, V. M., "First- and Second- Order Aerodynamic Sensitivity Derivatives Via Automatic Differentiation with Incremental Iterative Method," Proceedings of the Fifth AIAA/USAF/NASA/OAI Symposium on Multidisciplinary Analysis and Optimization, AIAA, Panama City, FL, September 1994, pp. 73–86. (AIAA Paper 94–4261–CP)

APPENDICES

APPENDIX A
GOVERNING EQUATIONS IN CURVILINEAR COORDINATES

The governing equations in the present study in three dimensions are the inviscid, compressible, unsteady Euler equations given in generalized curvilinear coordinates as follows:

$$\frac{\partial}{\partial t}\left(\widehat{Q}\right) + \frac{\partial}{\partial \xi}\left(\widehat{F}\right) + \frac{\partial}{\partial \eta}\left(\widehat{G}\right) + \frac{\partial}{\partial \zeta}\left(\widehat{H}\right) = 0 \tag{A.1}$$

where

$$\widehat{Q} = \frac{Q}{J} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ e \end{bmatrix}, \qquad \widehat{F} = \frac{F}{J} = \begin{bmatrix} \rho U \\ \rho Uu + \xi_x p \\ \rho Uv + \xi_y p \\ \rho Uw + \xi_z p \\ (e+p)U \end{bmatrix}$$

$$\widehat{G} = \frac{G}{J} = \begin{bmatrix} \rho V \\ \rho Vu + \eta_x p \\ \rho Vv + \eta_y p \\ \rho Vw + \eta_z p \\ (e+p)V \end{bmatrix}, \qquad \widehat{H} = \frac{H}{J} = \begin{bmatrix} \rho W \\ \rho Wu + \zeta_x p \\ \rho Wv + \zeta_y p \\ \rho Ww + \zeta_z p \\ (e+p)W \end{bmatrix}$$

$$U = \xi_x u + \xi_y v + \xi_z w$$

$$V = \eta_x u + \eta_y v + \eta_z w$$

$$W = \zeta_x u + \zeta_y v + \zeta_z w$$

above J is the Jacobian of the transformation from the Cartesian coordinates (x,y,z) to the generalized curvilinear coordinates (ξ,η,ζ), where ξ corresponds to the streamwise direction, η corresponds to the circumferential direction, and ζ corresponds to the direction normal to the body surface. The conservation laws of mass, and momentum in the X, Y, and Z directions and the energy equations are expressed symbolically in Eq. (A.1).

In the present study, the governing equations in two dimensions are the unsteady, compressible thin-layer Navier-Stokes equations given as

$$\frac{\partial}{\partial t}\left(\widehat{Q}\right) + \frac{\partial}{\partial \xi}\left(\widehat{F}\right) + \frac{\partial}{\partial \zeta}\left(\widehat{G} - \widehat{G}_v\right) = 0 \tag{A.2}$$

where

$$\widehat{Q} = \frac{Q}{J} = \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ e \end{bmatrix}, \qquad \widehat{F} = \frac{F}{J} = \begin{bmatrix} \rho U \\ \rho Uu + \xi_x p \\ \rho Uv + \xi_y p \\ (e+p)U \end{bmatrix}$$

$$\widehat{G} = \frac{G}{J} = \begin{bmatrix} \rho V \\ \rho Vu + \eta_x p \\ \rho Vv + \eta_y p \\ (e+p)V \end{bmatrix}, \qquad \widehat{G}_v^{tl} = \left(\frac{\mu}{Re_L}\right)\begin{bmatrix} \widehat{g}_{v_1} \\ \widehat{g}_{v_2} \\ \widehat{g}_{v_3} \\ \widehat{g}_{v_4} \end{bmatrix}$$

and

$$\widehat{g}_{v_1} = 0$$

$$\widehat{g}_{v_2} = \alpha_1 u_\zeta + \alpha_3 v_\zeta$$

$$\widehat{g}_{v_3} = \alpha_3 u_\zeta + \alpha_2 v_\zeta$$

$$\widehat{g}_{v_4} = \frac{1}{2}\alpha_1\left(u^2\right)_\zeta + \frac{1}{2}\alpha_2\left(v^2\right)_\zeta + \alpha_3(uv)_\zeta + \frac{\alpha_4}{Pr(\gamma-1)}\left(a^2\right)_\zeta$$

$$\alpha_1 = \left(\frac{\zeta_y^2}{J} + \frac{4}{3}\frac{\zeta_x^2}{J}\right), \qquad \alpha_2 = \left(\frac{\zeta_x^2}{J} + \frac{4}{3}\frac{\zeta_y^2}{J}\right)$$

$$\alpha_3 = \left(\frac{1}{3}\frac{\zeta_x\zeta_y}{J}\right), \qquad \alpha_4 = \left(\frac{\zeta_x^2 + \zeta_y^2}{J}\right)$$

The molecular viscosity is calculated with Stokes's hypothesis, a is the speed of sound, Pr is the Prandtl number ($Pr = 0.72$), and Re_L is the Reynolds number. The nondimensional molecular viscosity is calculated with Sutherland's law and a reference temperature T_∞, which is the static temperature of the free stream. For turbulent flow calculations, the algebraic turbulence model of Baldwin-Lomax is used to calculate the turbulent viscosity.

APPENDIX B
LINEARIZATION OF FAR-FIELD BOUNDARY
CONDITIONS FOR LIFTING AIRFOILS

The far-field boundary conditions used in this study are Riemann invariants. In this appendix, a procedure is outlined to linearize the far-field boundary conditions; this procedure is extended to include the lift-corrected far-field boundary condition.

The nonlinear residual expression on each boundary cell face can be written symbolically as

$$\{R_B(Q_B(\bar{\beta}), Q_{IP}(\bar{\beta}), \bar{X}_1(\bar{\beta}), \bar{\beta})\} = \{0\} \tag{B.1}$$

where $\{R_B\}$ is a four-component vector written as a function of the state variables on the boundary cell face Q_B, state variables at the first interior point Q_{IP}, local grid coordinates $\bar{X}_1$, and explicit dependence on the design variables $\bar{\beta}$. The two relationships enforced at each boundary cell face are given as follows (two components of $\{R_B\}$):

$$^1R \Rightarrow R_B^+ - R_{IP}^+$$
$$^2R \Rightarrow R_B^- - R_\infty^- \tag{B.2}$$

where ^1R is the outgoing Riemann invariant and ^2R is the incoming invariant. With these Riemann invariants, the local velocity $\bar{U}_B$ and the local speed of sound a_B are calculated as follows:

$$\bar{U}_B = \frac{^1R + \,^2R}{2}$$
$$a_B = \frac{(^1R - \,^2R)(\gamma - 1)}{4} \tag{B.3}$$

Based on the value of the local velocity $\bar{U}_B$, ^3R and ^4R (the third and fourth components of $\{R_B\}$) are enforced with the tangential velocity $\bar{V}$ and the entropy S as shown in Eqs. (B.4), where $\bar{U}_B > 0$ indicates the outgoing flow and $\bar{U}_B < 0$ indicates

$$
\begin{array}{ll}
\bar{U}_B > 0, & \bar{U}_B < 0 \\
^3R => \bar{V}_B - \bar{V}_{IP} = 0, & ^3R => \bar{V}_B - \bar{V}_\infty = 0 \\
^4R => S_B - S_{IP} = 0, & ^4R => S_B - S_\infty = 0
\end{array}
\qquad (B.4)
$$

the incoming flow. Here, the subscripts B, IP, and ∞ represent flow-field quantities on the boundary, on the first interior point, and for the free-stream, respectively.

By taking the derivative of the Eq. (B.1) with respect to the design variable β_k in the following equation results:

$$\left\{\frac{dR_B}{d\beta_k}\right\} = \{0\}$$

$$\left[\frac{\partial R_B}{\partial Q_B}\right]\left\{\frac{dQ_B}{d\beta_k}\right\} + \left[\frac{\partial R_B}{\partial Q_{IP}}\right]\left\{\frac{dQ_{IP}}{d\beta_k}\right\} + \left[\frac{\partial R_B}{\partial \bar{X}_l}\right]\left\{\frac{d\bar{X}_l}{d\beta_k}\right\} + \left\{\frac{\partial R_B}{\partial \beta_k}\right\} = \{0\} \quad (B.5)$$

where $\left[\frac{\partial R_B}{\partial Q_B}\right]$ and $\left[\frac{\partial R_B}{\partial Q_{IP}}\right]$ are 4×4 Jacobian matrices and $\left[\frac{\partial R_B}{\partial \bar{X}_l}\right]$ is a 4×2 Jacobian matrix. Here, the term $\left\{\frac{d\bar{X}_l}{d\beta_k}\right\}$ represents the grid-sensitivity vector. The vector $\left\{\frac{\partial R_B}{\partial \beta_k}\right\}$ is nonzero if the residual expression is explicitly dependent on the design variable β_k. Calculation of the expressions in Eq. (B.5) is straightforward and is not discussed here.

The lift-corrected far-field condition discussed in Ref. [61] has a distinct advantage because accurate force and moment coefficients can be calculated with a reduced extent of the far-field boundary. The use of the "point-vortex" correction to improve the far-field boundary condition is straightforward to implement in an explicit sense. Its explicit implementation involves the use of a point-vortex (centered at the quarter-chord) representation of the airfoil, where the strength of the point vortex (i.e., the circulation Γ) is proportional to the lift coefficient C_L of the airfoil. The purpose of this point vortex is to more accurately model the influence of the lifting airfoil on the velocity field in the

vicinity of the far-field boundaries (compared with the alternative of assuming a free-stream velocity field here), which results in more accurate airfoil calculations, particularly as the extent of the far-field boundary from the airfoil is decreased.

The implementation of this point-vortex correction results in a numerical coupling of the far-field boundary-condition equations to (through the lift coefficient C_L) the field variables and also to the (x, y) grid coordinates on and adjacent to the surface of the airfoil. As a consequence of this coupling between each far-field boundary condition equation and the field variables and grid points on and adjacent to the surface of the airfoil, algebraically, complex additions are necessary to the global Jacobian matrix $\left[\frac{\partial R}{\partial Q}\right]$ (which destroys the banded matrix structure) and also to $\left[\frac{\partial R}{\partial X}\right]$. To avoid the task of explicitly deriving these terms and their precise locations in these Jacobian matrices, a simplifying strategy is proposed.

Equation (B.1), with lift-coefficeint C_L as the additional field variable, is written as

$$\left\{ R_B\left(Q_B(\bar{\beta}), Q_{IP}(\bar{\beta}), \bar{X}_1(\bar{\beta}), \bar{\beta}, C_L\right) \right\} = \{0\} \tag{B.6}$$

The second and third components of Eq. (B.6) are different from Eq. (B.1), and the remaining two components of this four-component residual expression are the same. Only these two components are different because of the involvement of free-stream quantities, which are redefined with the lift-corrected far-field boundary condition. The free-stream quantities $\tilde{u}_\infty, \tilde{v}_\infty$, and $\tilde{a}_\infty$ are defined for the lift-corrected far-field boundary condition as

$$\tilde{u}_\infty = \cos\alpha + F\sin\theta$$

$$\tilde{v}_\infty = \sin\alpha - F\cos\theta$$

$$F = \frac{C_L \bullet C}{4\pi}\sqrt{1 - M_\infty^2}\,\frac{1}{\gamma[1 - M_\infty^2\sin(\theta - \alpha)]}$$

$$\tilde{a}_\infty = \sqrt{\left(h_{0\infty} - \frac{u^2 + v^2}{2}\right)(\gamma - 1)} \tag{B.7}$$

where r and θ are the radius and polar angle in the physical plane, M_∞ is the free-stream Mach number, α is the angle of attack, γ is the ideal gas constant, C is the chord of the airfoil, and $h_{0\infty}$ is the stagnation enthalpy. The polar angle is defined as positive counterclockwise from the chord line downstream of the airfoil quarter-chord. The speed of sound $\tilde{a}_\infty$ is determined by ensuring that the total enthalpy is constant. Here, the modified free-stream quantities are represented with '~'. The sine and cosine of the polar angle can be calculated as

$$\sin \theta = \frac{\Delta y}{r}$$
$$\cos \theta = \frac{\Delta x}{r}$$

where

$$\Delta x = x_p - x_0, \quad \Delta y = y_p - y_0$$
$$x_p = \frac{1}{2}(x_1 + x_2), \quad y_p = \frac{1}{2}(y_1 + y_2)$$
$$r = \sqrt{(\Delta x)^2 + (\Delta y)^2} \tag{B.8}$$

above, the quantities (x_0, y_0) represent the aerodynamic center of the airfoil. For the present study, $x_0 = C/4$ and $y_0 = 0$, where C is the chord of the airfoil. Quantities (x_p, y_p) represent the coordinates of a cell face, calculated by taking the average of the edges of the cell face. If we substitute for $\sin\theta$ and $\cos\theta$ in Eq (B.7) the following equation results:

$$\tilde{u}_\infty = \cos\alpha + C_L \Delta y \left(\frac{\sqrt{1 - M_\infty^2}}{4\pi} \right) f^{-1}$$
$$\tilde{v}_\infty = \sin\alpha + C_L(-\Delta x) \left(\frac{\sqrt{1 - M_\infty^2}}{4\pi} \right) f^{-1} \tag{B.9}$$

where

$$f = \left(1 - M_\infty^2 \sin^2\alpha\right)\triangle x^2 + \left(1 - M_\infty^2 \cos^2\alpha\right)\triangle y^2 + \left(2M_\infty^2 \sin\alpha\cos\alpha\right)\triangle x \triangle y$$

If we differentiate Eq. (B.6) with respect to the design variable β_k, the result is

$$\left[\frac{\partial R_B}{\partial Q_B}\right]\left\{\frac{dQ_B}{d\beta_k}\right\} + \left[\frac{\partial R_B}{\partial Q_{IP}}\right]\left\{\frac{dQ_{IP}}{d\beta_k}\right\} + \left[\frac{\partial R_B}{\partial \bar{X}_I}\right]\left\{\frac{d\bar{X}_I}{d\beta_k}\right\} + \left\{\frac{\partial R_B}{\partial \beta_k}\right\}$$

$$+ \left\{\frac{\partial R_B}{\partial C_L}\right\}\frac{dC_L}{d\beta_k} = \{0\} \tag{B.10}$$

The additional term in Eq. (B.10) compared with Eq. (B.5) is $\left\{\frac{\partial R_B}{\partial C_L}\right\}\frac{dC_L}{d\beta_k}$. The four-component vector $\left\{\frac{\partial R_B}{\partial C_L}\right\}$ can be easily computed because the explicit dependence of $\{R_B\}$ on C_L is known. The term $\frac{dC_L}{d\beta_k}$ is a scalar term that represents the sensitivity of the lift coefficient with respect to the design variable β_k. Throughout the remainder of this appendix, geometric design variables are discussed because the analytical expressions are not as straightforward to obtain in comparison with the expressions for the nongeometric design variables.

Here, the second and third components of Eq. (B.10) are discussed because of the complexity involved in calculating $\tilde{u}_\infty, \tilde{v}_\infty$, and $\tilde{a}_\infty$. The second component of Eq. (B.10) can be written as shown below:

$$\frac{d^2 R}{d\beta_k} = \frac{dR_{\bar{B}}}{d\beta_k} - \frac{dR_{\bar{\infty}}}{d\beta_k} \tag{B.11}$$

where

$$\frac{dR_{\bar{\infty}}}{d\beta_k} = \left[\tilde{u}_\infty M_1' + \tilde{v}_\infty M_2' + \left(M_1 + \frac{\tilde{u}_\infty}{\tilde{a}_\infty}\right)\tilde{u}_\infty' + \left(M_2 + \frac{\tilde{v}_\infty}{\tilde{a}_\infty}\right)\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\}$$

The derivative of $\frac{dR_{\bar{B}}}{d\beta_k}$ can be calculated analytically; the term $\frac{dR_{\bar{\infty}}}{d\beta_k}$ involves the metric terms M_1 and M_2 as well as the free-stream velocities $\tilde{u}_\infty$ and $\tilde{v}_\infty$ and their derivatives. These derivatives $\tilde{u}_\infty'$ and $\tilde{v}_\infty'$ are given as

$$\tilde{u}_\infty' = \left(\frac{C_L\sqrt{1 - M_\infty^2}}{4\pi}\right)\left[\frac{(\triangle y)'}{f} - \left(\frac{\triangle y}{f^2}\right)f'\right]$$

$$\tilde{v}'_\infty = \left(\frac{C_L\sqrt{1-M_\infty^2}}{4\pi}\right)\left[\frac{(-\Delta x)'}{f} + \left(\frac{\Delta x}{f^2}\right)f'\right]$$

$$(\Delta x)_y = 0, \qquad (\Delta y)_y = y'_p$$

$$(\Delta x)_x = y'_p, \qquad (\Delta y)_x = 0$$

$$f_x = (1 - M_\infty^2\sin^2\alpha)2\Delta x\Delta x' + 2M_\infty^2\sin\alpha\cos\alpha\,\Delta y\Delta x'$$

$$f_y = (1 - M_\infty^2\cos^2\alpha)2\Delta y\Delta y' + 2M_\infty^2\sin\alpha\cos\alpha\,\Delta x\Delta y' \qquad \text{(B.12)}$$

The derivatives of $\tilde{u}_\infty$ and $\tilde{v}_\infty$ with respect to x and y can be obtained by substituting the corresponding derivatives of Δx, Δy, and f as shown below. For example, the derivative of $\tilde{u}_\infty$ with respect to x can be shown as

$$\frac{\partial\tilde{u}_\infty}{\partial x} = \left(\frac{C_L\sqrt{1-M_\infty^2}}{4\pi}\right)\left(\frac{-\Delta y}{f^2}\right)[2\Delta x(1 - M_\infty^2\sin^2\alpha) + 2M_\infty^2\sin\alpha\cos\alpha\,\Delta y]x'_p$$

$$\text{(B.13)}$$

where the derivatives of f and Δy with respect to x are substituted in the expression for $\tilde{u}'_\infty$.

The third component of Eq. (B.10) can be written as

$$^3R => \bar{V}_B - \bar{V}_\infty = 0 \qquad \text{(B.14)}$$

where $\bar{V}_B$ and $\bar{V}_\infty$ are tangential velocities on the boundary and at the free stream. The velocities $\bar{V}_B$ and $\bar{V}_\infty$ can be calculated as

$$\bar{V}_B = M_2 u_B - M_1 v_B$$

$$\bar{V}_\infty = M_2\tilde{u}_\infty - M_1\tilde{v}_\infty \qquad \text{(B.15)}$$

where M_1 and M_2 are metric terms and u_B and v_B are the Cartesian components of velocity on the boundary cell face.

If we differentiate Eq. (B.14) with respect to the design variable β_k, then the following equation results:

$$\frac{d^3 R}{d\beta_k} = \frac{d\bar{V}_B}{d\beta_k} - \frac{d\bar{V}_\infty}{d\beta_k}$$

(B.16)

In Eq. (B.16), the term $\frac{d\bar{V}_B}{d\beta_k}$ is straightforward to obtain. Derivatives of $\bar{V}_\infty$ with respect to the design variable β_k can be obtained by differentiating the expression for $\bar{V}_\infty$ from Eq. (B.15), where the derivative quantities $\tilde{u}'_\infty$ and $\tilde{v}'_\infty$ are calculated as shown in Eq. (B.12).

APPENDIX C
ADJOINT VARIABLE FORMULATION FOR MARCHING
EULER PROBLEMS IN THREE DIMENSIONS

In this appendix, the adjoint-variable approach to calculate SD's is outlined for the

Euler equations in three dimensions with a space marching algorithm. This procedure

has not yet been implemented in the present study. The system response C is augmented

with the product of the Lagrangian multiplier λ_i and the residual R_i (where i corresponds

to the ith cross plane in the streamwise direction) as

$$C = C + \lambda_i^T R_i(Q_i^*(\bar{\beta}), Q_{i-1}^*(\bar{\beta}), Q_{i-2}^*(\bar{\beta}), \bar{X}, \bar{\beta}) \qquad (C.1)$$

At steady state, R_i clearly is equal to zero. Here, Q_i^*, Q_{i-1}^*, and Q_{i-2}^* represent the

steady-state field variables in the i, $i - 1$ and $i - 2$ cross planes, respectively, and the j

and k indices are suppressed. If we differentiate Eq. (C.1) with respect to the design

variable β_k, the following equation results:

$$\frac{dC}{d\beta_k} = \left\{\frac{\partial C}{\partial Q_1}\right\}^T \left\{\frac{dQ_1^*}{d\beta_k}\right\} \cdots + \left\{\frac{\partial C}{\partial Q_i}\right\}^T \left\{\frac{dQ_i^*}{d\beta_k}\right\} \cdots + \left\{\frac{\partial C}{\partial Q_{imax}}\right\}^T \left\{\frac{dQ_{imax}^*}{d\beta_k}\right\}$$

$$+ \left\{\frac{\partial C}{\partial \bar{X}}\right\}^T \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C}{\partial \beta_k}$$

$$+ \lambda_1^T \left(\left[\frac{\partial R_1}{\partial Q_1}\right]\left\{\frac{dQ_1^*}{d\beta_k}\right\} + \left[\frac{\partial R_1}{\partial Q_0}\right]\left\{\frac{dQ_0}{d\beta_k}\right\} + \left[\frac{\partial R_1}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_1}{\partial \beta_k}\right\}\right)$$

$$+ \cdots\cdots\cdots\cdots\cdots\cdots$$

$$+ \lambda_i^T \left(\left[\frac{\partial R_i}{\partial Q_i}\right]\left\{\frac{dQ_i^*}{d\beta_k}\right\} + \left[\frac{\partial R_i}{\partial Q_{i-1}}\right]\left\{\frac{dQ_{i-1}^*}{d\beta_k}\right\} + \left[\frac{\partial R_i}{\partial Q_{i-2}}\right]\left\{\frac{dQ_{i-2}^*}{d\beta_k}\right\}\right)$$

$$+ \lambda_i^T \left(\left[\frac{\partial R_i}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_i}{\partial \beta_k}\right\}\right)$$

$$+ \cdots\cdots\cdots\cdots\cdots\cdots$$

$$+ \lambda_{imax}^T \left(\left[\frac{\partial R_{imax}}{\partial Q_{imax}}\right]\left\{\frac{dQ_{imax}^*}{d\beta_k}\right\} + \left[\frac{\partial R_{imax}}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_{imax}}{\partial \beta_k}\right\}\right) \qquad (C.2)$$

In Eq. (C.2), the terms that correspond to the first cross plane, the ith cross plane, and the *imax* cross plane (*imax* is the number cross planes in the i direction) are given; the reason for showing these terms in the equation becomes clear later in this appendix. The Jacobian matrix $\left[\frac{\partial R_i}{\partial X}\right]$ is a sparse, banded matrix and is calculated as $\left[\frac{\partial R_i}{\partial M}\right]\left[\frac{\partial M}{\partial X}\right]$, where M is the metric term. The derivative of the residual expression with respect to the metric terms is straightforward and is not given here. More details in regard to the construction of this Jacobian matrix are given in Ref. [35]. Contributions from boundary conditions are included in the above Jacobian matrix, which are essential for calculating accurate SD's. The term $\left\{\frac{d\bar{X}}{d\beta_k}\right\}$ is the grid-sensitivity vector, which is discussed in detail in Chap. 3. As can be seen from Eq. (C.2), necessary adjustments are needed when $i = 1$ and the flow variables that correspond to the free stream are used for Q_0. In Eq. (C.2), $\left[\frac{\partial R_i}{\partial Q_i}\right]$ is the implicit Jacobian matrix discussed in Chap. 2. The term $\left[\frac{\partial R_i}{\partial X}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\}$ is nonzero if the design variable is geometric, and the term $\left\{\frac{\partial R_i}{\partial\beta_k}\right\}$ is nonzero if the design variable is nongeometric. By rearranging Eq. (C.2) and collecting terms that correspond to the sensitivity of the flow variables, we obtain

$$
\begin{aligned}
\frac{dC}{d\beta_k} &= \left\{\frac{\partial C}{\partial \bar{X}}\right\}^T \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C}{\partial \beta_k} + \sum_{i=1}^{imax} \lambda_i^T \left(\left[\frac{\partial R_i}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_i}{\partial \beta_k}\right\}\right) \\
&+ \left\{\frac{dQ_1^*}{d\beta_k}\right\}\left(\lambda_1^T\left[\frac{\partial R_1}{\partial Q_1}\right] + \lambda_2^T\left[\frac{\partial R_2}{\partial Q_1}\right] + \lambda_3^T\left[\frac{\partial R_3}{\partial Q_1}\right] + \left\{\frac{\partial C}{\partial Q_1}\right\}^T\right) \\
&+ \ldots\ldots\ldots\ldots \\
&+ \left\{\frac{dQ_i^*}{d\beta_k}\right\}\left(\lambda_i^T\left[\frac{\partial R_i}{\partial Q_i}\right] + \lambda_{i+1}^T\left[\frac{\partial R_{i+1}}{\partial Q_i}\right] + \lambda_{i+2}^T\left[\frac{\partial R_{i+2}}{\partial Q_i}\right] + \left\{\frac{\partial C}{\partial Q_i}\right\}^T\right) \\
&+ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\
&+ \left\{\frac{dQ_{imax}^*}{d\beta_k}\right\}\left(\lambda_{imax}^T\left[\frac{\partial R_{imax}}{\partial Q_{imax}}\right] + \left\{\frac{\partial C}{\partial Q_{imax}}\right\}^T\right)
\end{aligned} \tag{C.3}
$$

In Eq. (C.3), if we set the coefficients of $\left\{\frac{dQ_i^*}{d\beta_k}\right\}$ to zero, the following equation results:

$$
\frac{dC}{d\beta_k} = \left\{\frac{\partial C}{\partial \bar{X}}\right\}^T \left\{\frac{d\bar{X}}{d\beta_k}\right\} + \frac{\partial C}{\partial \beta_k} + \sum_{i=1}^{imax} \lambda_i^T \left(\left[\frac{\partial R_i}{\partial \bar{X}}\right]\left\{\frac{d\bar{X}}{d\beta_k}\right\} + \left\{\frac{\partial R_i}{\partial \beta_k}\right\}\right) \tag{C.4}
$$

where the adjoint vectors are solved with Eq. (C.5).

$$-\left[\frac{\partial R_1}{\partial Q_1}\right]^T \lambda_1 = \left[\frac{\partial R_2}{\partial Q_1}\right]^T \lambda_2 + \left[\frac{\partial R_3}{\partial Q_1}\right]^T \lambda_3 + \left\{\frac{\partial C}{\partial Q_1}\right\}$$

$$\ldots\ldots$$

$$-\left[\frac{\partial R_i}{\partial Q_i}\right]^T \lambda_i = \left[\frac{\partial R_{i+1}}{\partial Q_i}\right]^T \lambda_{i+1} + \left[\frac{\partial R_{i+2}}{\partial Q_i}\right]^T \lambda_{i+2} + \left\{\frac{\partial C}{\partial Q_i}\right\}$$

$$\ldots\ldots$$

$$-\left[\frac{\partial R_{imax}}{\partial Q_{imax}}\right]^T \lambda_{imax} = \left\{\frac{\partial C}{\partial Q_{imax}}\right\} \tag{C.5}$$

As can be seen from Eq. (C.5), we must solve for the adjoint vectors backwards (i.e., we solve for λ_{imax} first and use it to solve for λ_{imax-1} and so on). Equation (C.5) can be cast in incremental form. The incremental form to solve for λ_i is given as a two-step procedure in Eqs. (C.6a) and (C.6b):

$$-\left[\widehat{\frac{\partial R_i}{\partial Q_i}}\right]^T \Delta\lambda_i = \left[\frac{\partial R_i}{\partial Q_i}\right]^T \lambda_i^m + \left[\frac{\partial R_{i+1}}{\partial Q_i}\right]^T \lambda_{i+1} + \left[\frac{\partial R_{i+2}}{\partial Q_i}\right]^T \lambda_{i+2} + \left\{\frac{\partial C}{\partial Q_i}\right\} \tag{C.6a}$$

$$\left\{\lambda_i^{m+1}\right\} = \left\{\lambda_i^m\right\} + \left\{{}^m\Delta\lambda_i\right\}$$

$$m = 1, 2, 3\ldots \tag{C.6b}$$

APPENDIX D
WING-GEOMETRY PARAMETERIZATION

The baseline HSCT 24E wing-geometry parameterization of Ref. [81] was divided into three types: 7 planform variables, 15 section-thickness variables (5 each at the root, break, and tip section), and 20 camber surface variables. These camber surface elevation variables were simply the coefficients in a monomial product expansion of 20 terms, such as $a\left(\frac{X}{C}\right)^n \left(\frac{2y}{b}\right)^m$. In the present work, the camber parameterization has been changed from that shown in Ref. [81]; however, the parameterization for the planform and thickness variables have been retained.

The HSCT 24E geometry generated at NASA Langley Research Center resulted from a multidisciplinary preliminary design based on linear aerodynamic codes; the geometry is given in the wave-drag format. The wing is described at 18 span stations, which are located as shown in Table D1. The seven planform variables required to describe the double trapezoidal wing used in Ref. [81] are defined in Table D2 and Fig. D1. The inboard- and outboard-span variables are shown with dashed arrows because they are not involved in any present optimization studies. Because the HSCT 24E wing-thickness distribution was linearly lofted from root to break and from break to tip, a thickness parameterization is required only at these three wing stations. The thickness parameterization used in Ref. [81] and in this work is defined in Table D3.

The HSCT 24E wing camber surface is described in the wave-drag, or Harris, format by 20 chordwise entries at each of the 18 span stations (i.e., 360 parameters). In the present work, the camber has been described at each wing station so that twist and both leading- and trailing-edge flaps can be included. Locations of the four outboard flaps on the HSCT 24E are shown in Fig. D2. The twist, camber, and flap parameterizations

are defined and shown in Table D4, Fig. D3, and Fig. D4. This present parameterization requires the 72 (18×4) camber variables to approximate the HSCT 24E wing camber surface elevation; this representation is better than that obtained with the representation with 20 camber variables given in Ref. [81]. Additional spanwise control (or smoothing) is required to model the flaps and for the optimization design-variable changes discussed in the text.

Table D1 HSCT 24E Wing-Section Locations

Wing section	% distance along the span from side of fuselage
1 (Root)	0.00
2	5.94
3	11.88
4	17.82
5	23.77
6	29.71
7	35.65
8 (Break)	42.44
9	47.53
10	53.47
11	59.42
12	65.36
13	71.30
14	77.24
15	83.18
16	89.12
17	95.06
18 (Tip)	100.00

Table D2 Planform Parameters

RC	Root chord
BC	Break chord
TC	Tip chord
XBC	X - location of leading edge at break
XTC	X - location of leading edge at tip
IS	Inboard span
OS	Outboard span

Table D3 Thickness Parameters

I	Leading-edge radius parameter, $R_0 = 1.1019 * [(I/6.0) * *2]$
B	Curvature forward of airfoil maximum thickness
t/C	Thickness to chord ratio
Xm	Location in (x/C) of airfoil maximum thickness
TAU	Thickness trailing-edge half-angle

Table D4 Camber and Flap Parameters

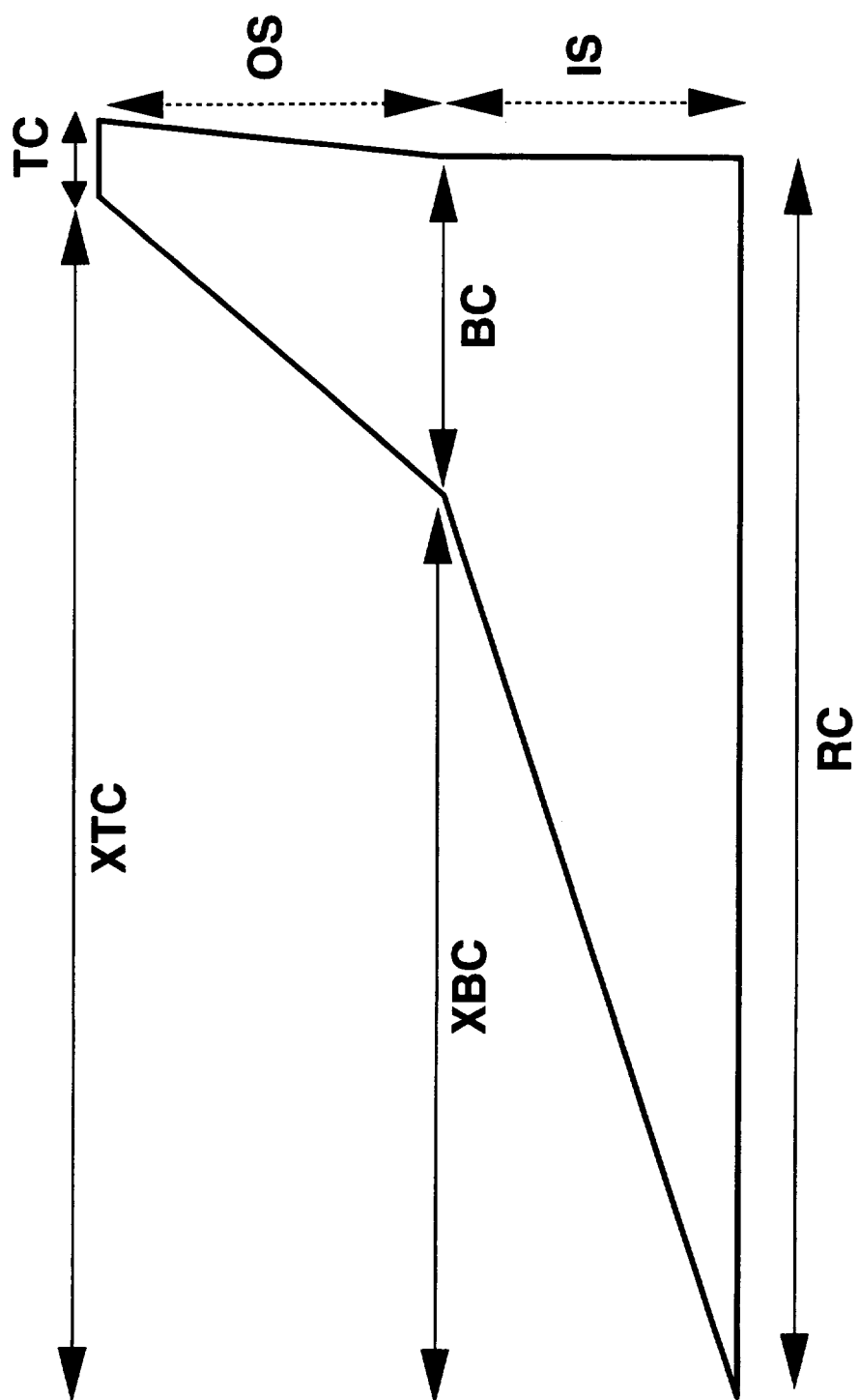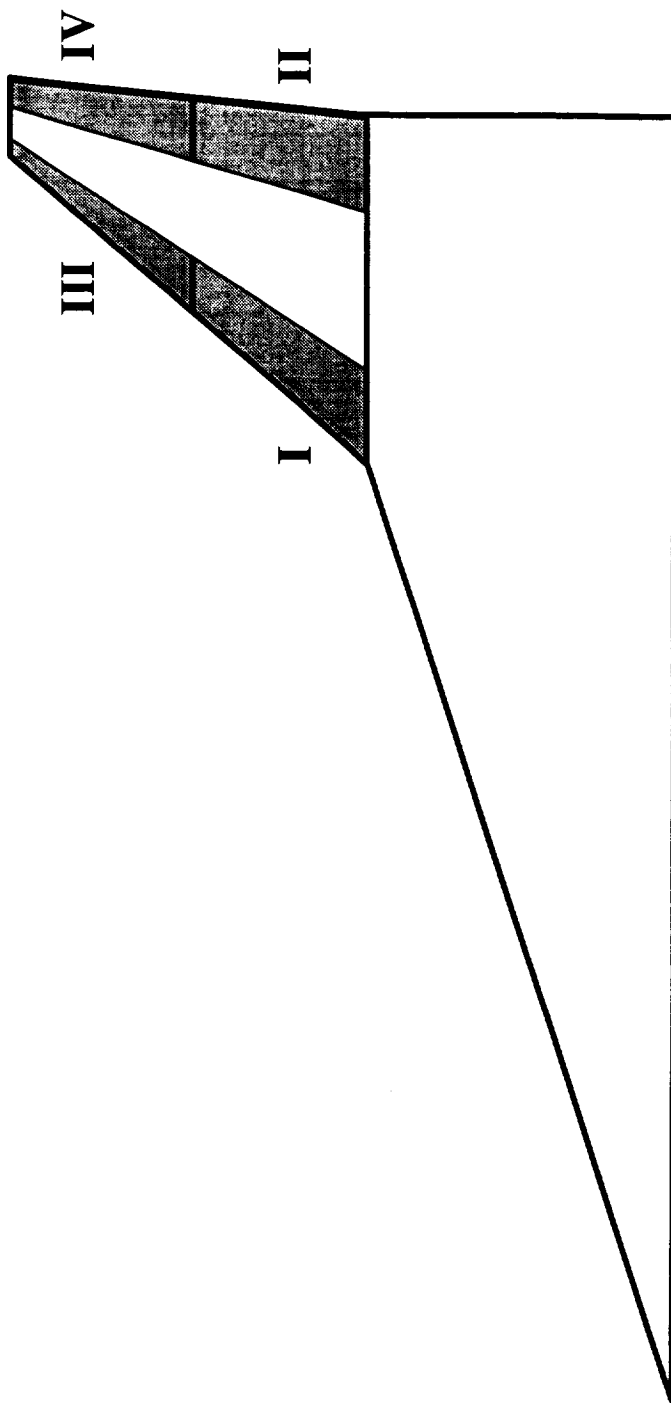|	
ZTE	Twist
A	Camber
E	Camber inflection
XMA	X/C location of maximum camber
XHL	X/C location of leading-edge flap hinge
θ_L	Deflection of leading-edge flap
XHT	X/C location of trailing-edge flap hinge
θ_T	Deflection of trailing-edge flap

Fig. D1 Wing-planform parameterization.

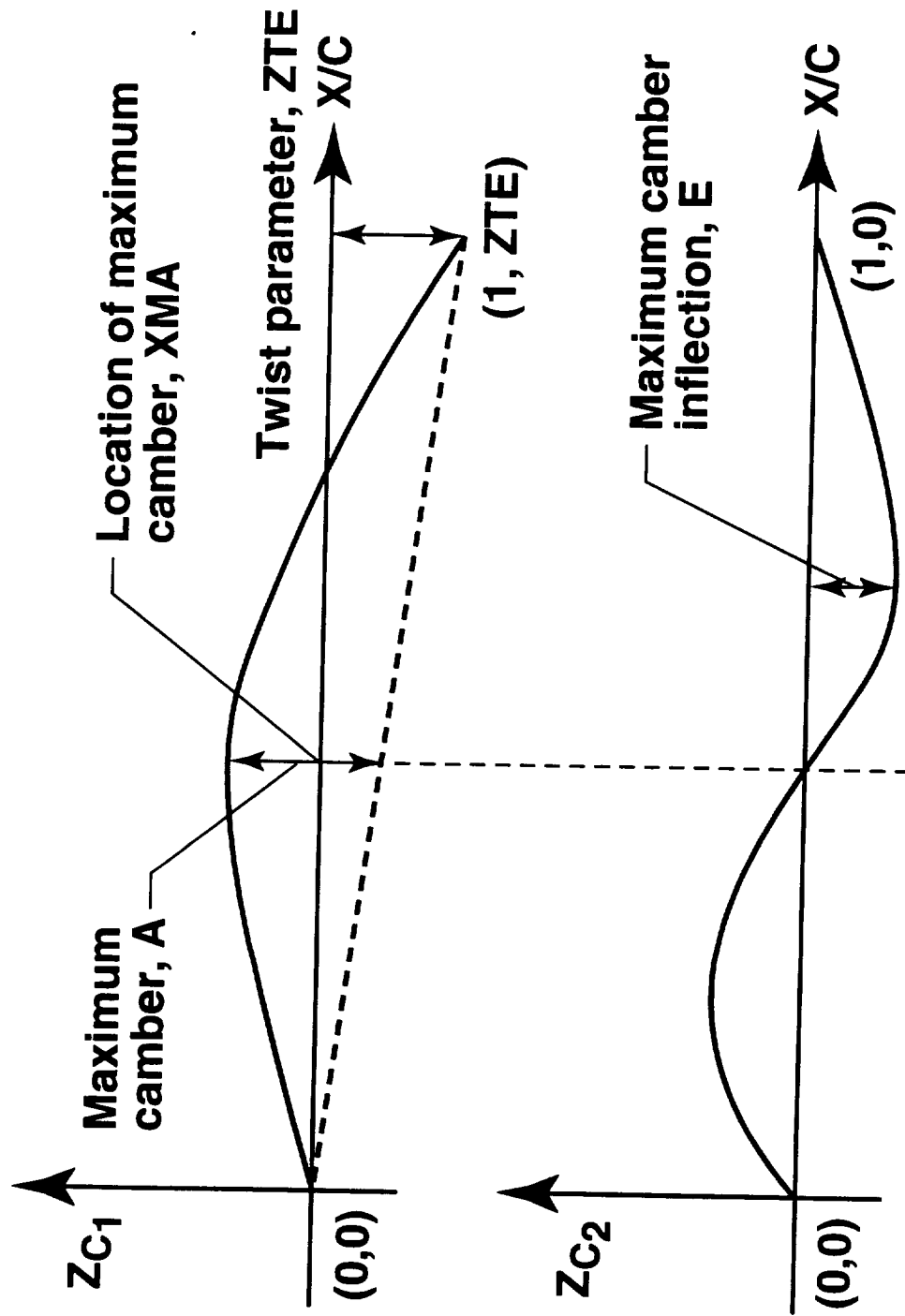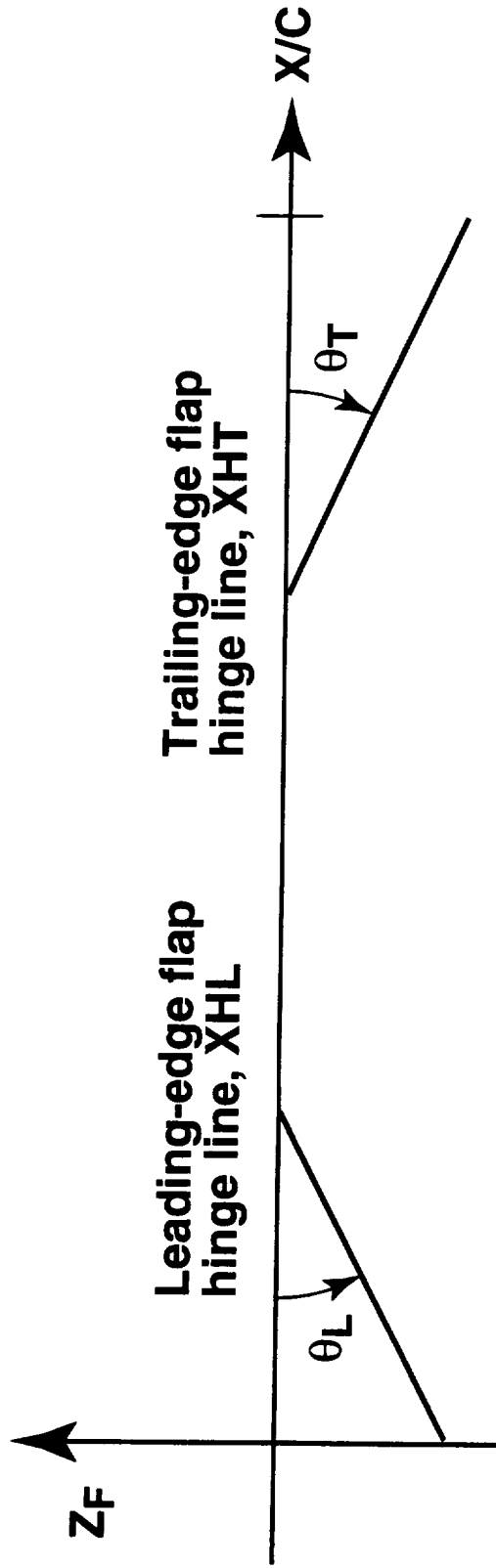Fig. D2 Outboard wing flap locations for HSCT 24E.

Fig. D3 Wing-section camber parameterization: Twist and camber.

$$Z_{C_1} = ZTE * (X/C) + (A - ZTE) * (X/C) * [(X/C) - 1]$$

$$Z_{C_2} = E * (X/C) * [(X/C) - 1] * [(X/C) - (XMA/C)]$$

$$Z_C = Z_{C_1} + Z_{C_2} + Z_F$$

Fig. D4 Wing-section camber parameterization: Flaps.

APPENDIX E
AUTOMATIC DIFFERENTIATION

An AD tool is a chain-rule-based technique for differentiating an output function of a program with respect to some specified input parameters. This technique is as old as programmable systems [84]. This AD tool relies on the technique that every function is calculated on a computer by executing some basic operations such as addition, subtraction, and multiplication. Principally, two modes exist in automatic differentiation: the forward mode and the reverse mode (which closely resembles the adjoint approach with a low operation count and a large computational memory requirement).

An AD tool computes derivatives within the accuracy of the original function, unlike divided differences. These tools differ from a symbolic manipulator in that the operation count and memory are bounded *a priori* in terms of the complexity of the original code. Calculation of the SD's by hand differentiation is not feasible for complicated CFD codes. For example, the differentiation of turbulence models by hand-differentiation is not feasible, and failure to consistently differentiate these terms results in inaccurate SD's as shown by Korivi et al. [42]. Hand differentiation is error prone and requires a lot of time to construct the differentiation code; on the other hand, automatic differentiation constructs accurate derivatives of very complex codes in a very short time. In the near future, usage of these codes may become routine for computing derivatives accurately and efficiently; this tool can be used judiciously to obtain SD's. (The case of using an AD tool to obtain SD via the IIM is discussed later.)

The AD source tool used in the present study, ADIFOR (Automatic DIfferentiation of FORTRAN) [86–88], is jointly developed by Argonne National Laboratories and Rice University. The ADIFOR tool differentiates any specified FORTRAN program

output with respect to any program input parameters and uses a hybrid mode of forward and reverse modes of AD; ADIFOR is a general-purpose tool that supports almost all of FORTRAN 77 and is based on the ParaScope FORTRAN environment. The differentiation of a FORTRAN program output with respect to an input parameter using ADIFOR produces a FORTRAN code that computes the derivative of the function and also computes the function itself upon execution of the resultant code. The original program vectorization and parallelization are preserved and supports the error exception handling routines. The Jacobian matrix is computed with the low-memory-based seed matrix concept. The number of columns in the seed-matrix is the number of design variables. More details in regard to how ADIFOR handles sparsity are given in Ref. [87].

The ADIFOR tool has been applied to various Fortran codes to obtain SD's from advanced CFD codes. Bischof et al. [89] and Green et al. [90] applied ADIFOR to TLNS3D to obtain accurate SD's with respect to turbulence modeling parameters and nongeometric design variables. The application of ADIFOR to an iterative algorithm is demonstrated in these studies. The application of ADIFOR to an iterative procedure such as

$$X^{n+1} = X^n - P^{-1} * R \qquad (E.1)$$

(which is a common iterative procedure in any CFD code, where P is the preconditioner, R is the residual, and n is the iteration index) results in the following iterative procedure:

$$X'^{n+1} = X'^n - \left(P^{-1}\right)' * R - P^{-1} * R' \qquad (E.2)$$

where the derivative of the preconditioner $\left(P^{-1}\right)'$ is also calculated. This iterative procedure is used to compute derivatives from a differentiated version of TLNS3D. However, in Eq. (E.2) the derivative of the preconditioner is computed and multiplied by the residual at each iteration. This can be avoided because R is equal to zero at steady state. Bischof et al. [89] suggested the deactivation of certain parts of the differentiated program to calculate the derivatives. This step needs user intervention

and is not automatic. Newman et al. [43] suggested that the use of ADIFOR with the IIM results in an accurate and efficient evaluation of derivatives where only the derivative of the residual is computed. The preconditioner used for the SD evaluation is the same as that used for the analysis. Sherman et al. [91] applied ADIFOR via the IIM to compute first- and second-order derivatives from a Navier-Stokes code with an algebraic turbulence model. The SD's computed with respect to geometric and nongeometric design variables compare well with those computed with finite differences. Korivi et al. [60] and Green et al. [77] applied ADIFOR to an algebraic grid-generation code to compute the grid sensitivity and successfully obtained the SD's with respect to the geometric design variables.